Élan™SC400 and ÉlanSC410

Single-Chip, Low-Power, PC/AT-Compatible Microcontrollers

DISTINCTIVE CHARACTERISTICS

Élan™SC400 and ÉlanSC410 Microcontrollers

- **E86™ family of x86 embedded processors**
  - Offers improved time-to-market, software migration, and field-proven development tools
- Highly integrated single-chip CPU with a complete set of common peripherals
  - Accelerates time-to-market with simplified hardware
  - Low-power 0.35-micron process technology
  - Single chip delivers smallest system form factor
  - 33-MHz, 66-MHz, and 100-MHz operating frequencies
- **Am486® CPU core**
  - Robust Microsoft® Windows® compatible CPU
  - 8-Kbyte write-back cache for enhanced performance
  - Fully static design with System Management Mode (SMM) for power savings
- **Comprehensive power management unit**
  - Seven modes of operation allow fine-tuning of power requirements for maximum battery life
  - Provides a superset of APM 1.2 features
- **Glueless burst-mode ROM/Flash memory/SRAM interface**
  - Reduces system cost by allowing mask ROM and Flash memory at the same time with three ROM/Flash memory/SRAM chip selects
- **Glueless DRAM controller**
  - Allows mixed DRAM types on a per-bank basis to reduce system cost
- **VESA Local (VL) bus and ISA bus interface**
  - Reduces time-to-market with a wide variety of off-the-shelf companion chips
- **Standard PC/AT system logic (PICs, DMACs, timer, RTC)**
  - DOS, ROM-DOS, Windows, and industry-standard BIOS support
  - Leverages the benefits of desktop computing environment at embedded price points
- **Bidirectional parallel port with Enhanced Parallel Port (EPP) mode**
- **16550-compatible UART**
- **Infrared port for wireless communication**
  - Standard and high-speed
- **Keyboard interface**
  - Matrix keyboard support with up to 15 rows and 8 columns
  - SCP-emulation mode for PC/AT and XT keyboard support

ÉlanSC400 Microcontroller Only

The ÉlanSC400 microcontroller includes the following additional features designed specifically for mobile computing applications. The ÉlanSC410 microcontroller does not include these features.

- **Dual PC Card (PCMCIA Version 2.1) controller supports 8- or 16-bit data bus**
  - End-user (after-market) system expansion
  - ExCA-compliant, 82365-register set compatible
  - Leverages off-the-shelf card and socket services
  - Supports DMA transfers between I/O PC cards and system DRAM
- **LCD graphics controller**
  - Supports monochrome and 4-bit color Super Twisted Nematic (STN) LCDs
  - Unified Memory Architecture (UMA) eliminates separate video memory

Publication# **21028** Rev: **B** Amendment/**0**
Issue Date: **December 1998**

GENERAL DESCRIPTION

The Élan™SC400 and ÉlanSC410 microcontrollers are the among the latest in a series of E86™ family microcontrollers, which integrate proven x86 CPU cores with a comprehensive set of on-chip peripherals in a 0.35-micron process.

The ÉlanSC400 and ÉlanSC410 microcontrollers combine a 32-bit, low-voltage Am486 CPU with a complete set of PC/AT-compatible peripherals, along with the power management features required for battery operation.

Leveraging the benefits of the x86 desktop computing environment, the ÉlanSC400 and ÉlanSC410 microcontrollers integrate all of the common logic and I/O functionality associated with a PC/AT computing system into a single device, eliminating the need for multiple peripheral chips. Fully integrated PC/AT-compatible peripherals include two 8259A-compatible programmable interrupt controllers (PICs), two 8237A-compatible DMA controllers, an 8254-compatible timer, a 16550 UART, an IrDA controller, VL-bus and ISA bus controllers, a real-time clock (RTC), and Enhanced Parallel Port (EPP) mode for the parallel port.

With its low-voltage Am486® CPU core and ultra-small form factor, the ÉlanSC400 microcontroller is highly optimized for mobile computing applications. The ÉlanSC410 microcontroller is targeted specifically for embedded systems.

A feature comparison of the two microcontrollers is shown in Table 1 on page 3.

The ÉlanSC400 and ÉlanSC410 microcontrollers use the industry-standard 486 microprocessor instruction set. All software written for the x86 architecture family is compatible with the ÉlanSC400 and ÉlanSC410 microcontrollers.

The ÉlanSC400 and ÉlanSC410 microcontrollers are based on a fully static design and include an advanced power management unit. Operating voltages are 2.7 V–3.3 V with 5-V-tolerant I/O pads. Orderable in both 33-MHz, 66-MHz, and 100-MHz peak processor speeds, the product is available in the ultra-small 292 ball grid array (BGA) package.

ORDERING INFORMATION

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of the elements below.

ELANSC400 –33 A C

TEMPERATURE RANGE
C = Commercial
For 33 and 66 MHz: T_{CASE} = 0°C to +95°C
For 100 MHz: T_{CASE} = 0°C to +85°C

I = Industrial
For 33 and 66 MHz, T_{CASE} = – 40°C to +95°C

PACKAGE TYPE
A = 292-pin BGA (Ball Grid Array)

SPEED OPTION
–33 = 33 MHz
–66 = 66 MHz
–100 = 100 MHz

DEVICE NUMBER/DESCRIPTION
ÉlanSC400 microcontroller
ÉlanSC410 microcontroller

Valid Combinations	
ELANSC400–33	AC, AI
ELANSC400–66	
ELANSC400–100	AC
ELANSC410–33	AC, AI
ELANSC410–66	
ELANSC410–100	AC

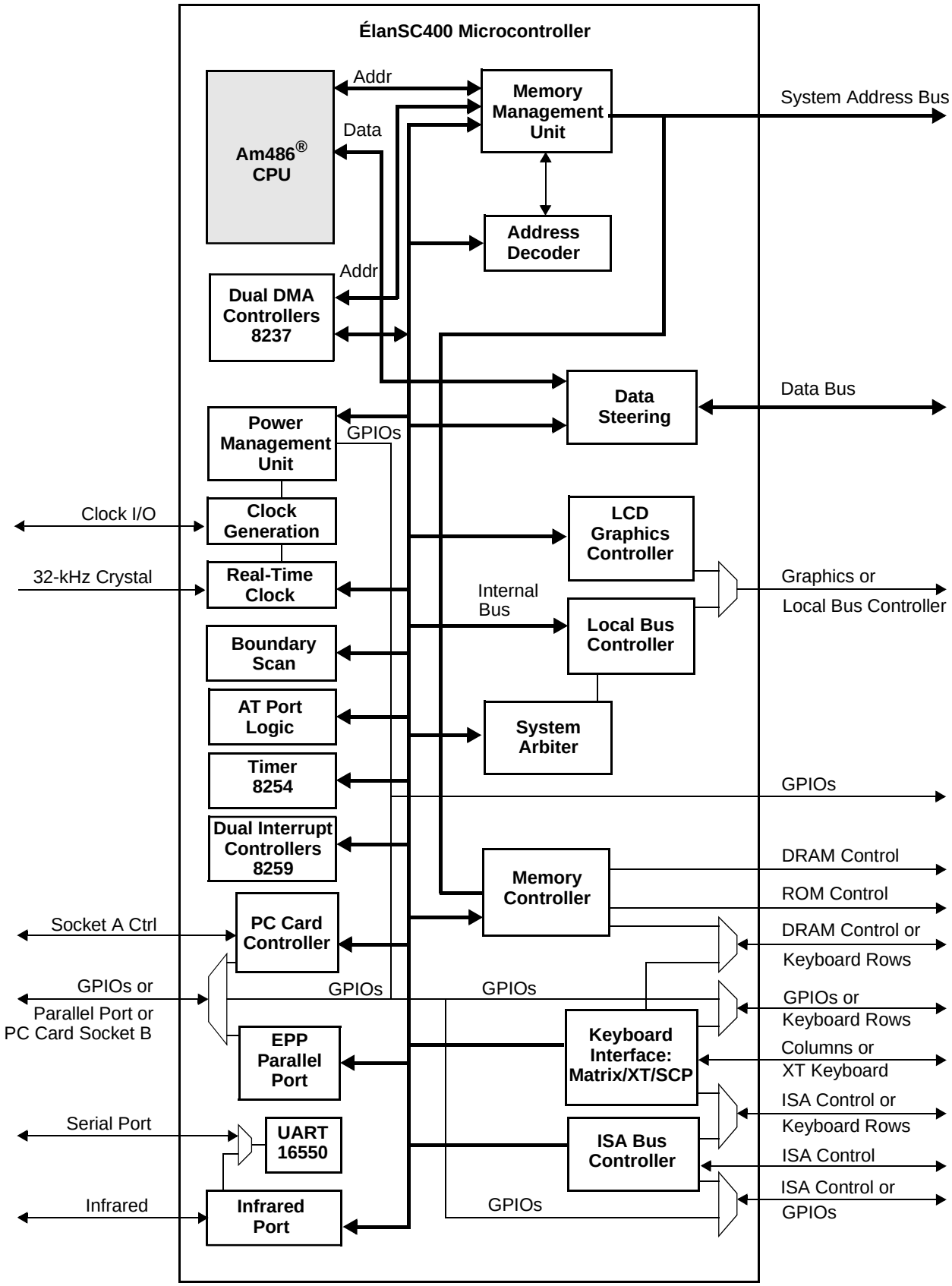
Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations and to check on newly released combinations.

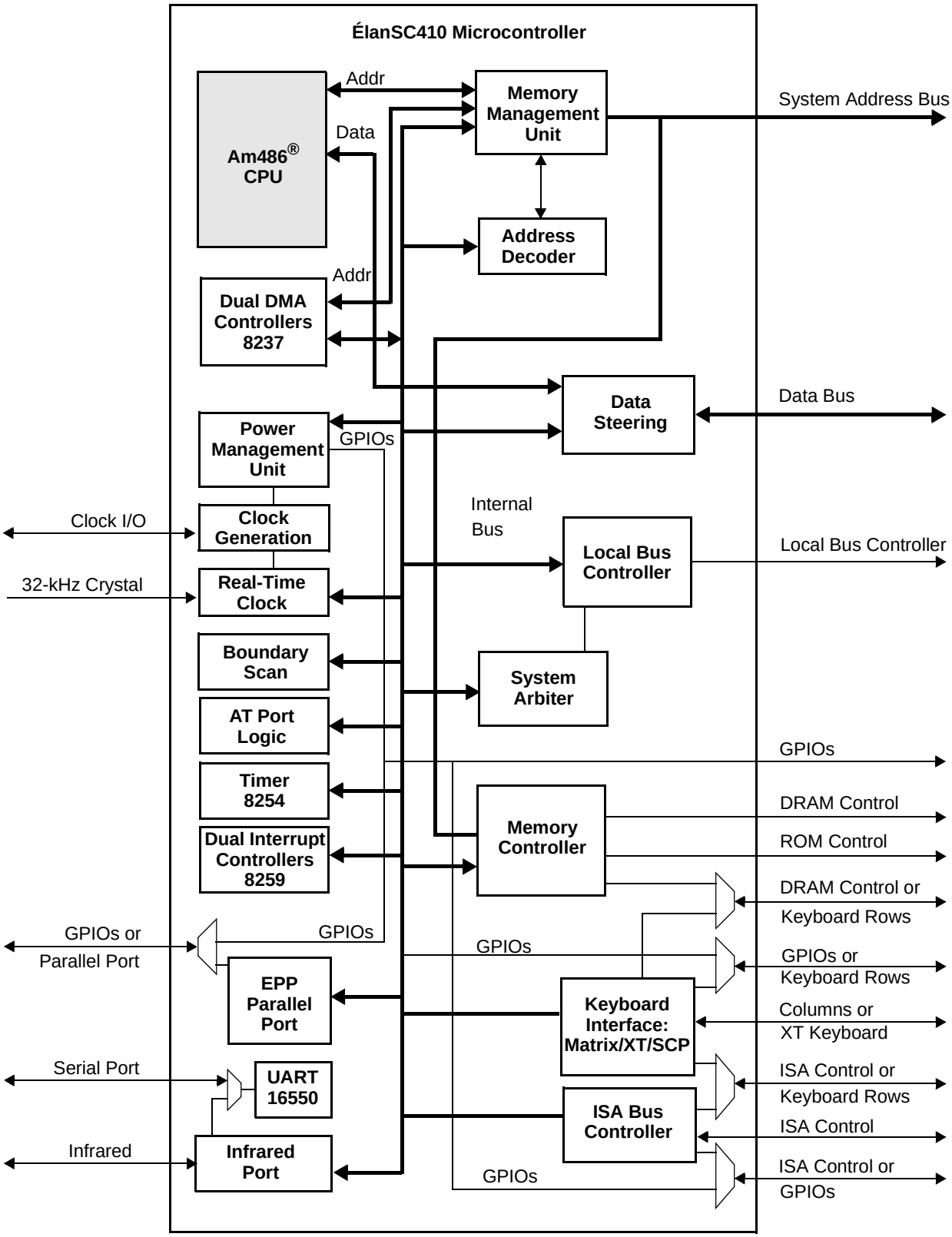
Table 1. Product Comparison—ÉlanSC400 and ÉlanSC410 Microcontrollers

Feature	ÉlanSC410	ÉlanSC400
Core CPU	Am486 CPU	Am486 CPU
L1 Cache	8-Kbyte Write-Back	8-Kbyte Write-Back
System management mode (SMM)	Yes	Yes
Floating-point unit	No	No
Data Bus	16, 32 bit	16, 32 bit
ISA Interface	8, 16 bit	8, 16 bit
ISA bus mastering	No	No
VESA Local Bus	32 bit	32 bit
VL bus mastering	No	No
Power Management	Yes	Yes
Mode timers	Yes	Yes
Activity detection	Yes	Yes
SMI/NMI generation	Yes	Yes
Battery monitoring	Yes	Yes
On-Chip ROM Interface		
Width	8, 16, 32 bit	8, 16, 32 bit
Size (total ROM space)	3 x 64 Mbyte	3 x 64 Mbyte
ROM chip selects	3	3
Burst-mode support	Yes	Yes
Support for SRAM as ROM address space	Yes	Yes
On-Chip DRAM Controller		
Banks	4	4
Width	16, 32 bit	16, 32 bit
Size (total of all banks)	64 Mbyte	64 Mbyte
EDO support	Yes	Yes
Support for SRAM as main memory	ROM-mappable	ROM-mappable
Integrated PC/AT-Compatible Peripherals		
Programmable timer (8254-compatible)	Yes	Yes
Real-time clock (146818A-compatible)	Yes	Yes
Port B and Port 92h I/O registers	Yes	Yes
Cascaded DMA Controllers (8237A)	2	2
Width	8, 16 bit	8, 16 bit
Total number of channels	7	7
External channels	2	2
Cascaded Interrupt Controllers (8259)	2	2
External IRQ signals	8	8
Bidirectional Parallel Port with EPP Mode	Yes	Yes
Serial Port (UART)	16550-compatible	16550-compatible
Keyboard Interface		
Support for external 8042 SCP	Yes	Yes
XT interface	Yes	Yes
Matrix scanned with SCP emulation	Yes	Yes
General-Purpose Input/Output Signals	32	32
Infrared (IrDA) Port	Yes	Yes
PC Card Controller	No	Yes
Sockets		2
PCMCIA 2.1-compliant		Yes
82365-compatible		Yes
LCD Graphics Controller	No	Yes
Programmable clock frequency		Yes
Unified memory architecture (UMA)		Yes
JTAG Support	Yes	Yes
Pin Count and Package	292 BGA	292 BGA
V_{CC}: CPU core	2.7–3.3 V	2.7–3.3 V
On-chip peripheral logic	3.3 V	3.3 V
I/O tolerance (designated pins)	5 V	5 V
Processor Clock Rate	33, 66, 100 MHz	33, 66, 100 MHz

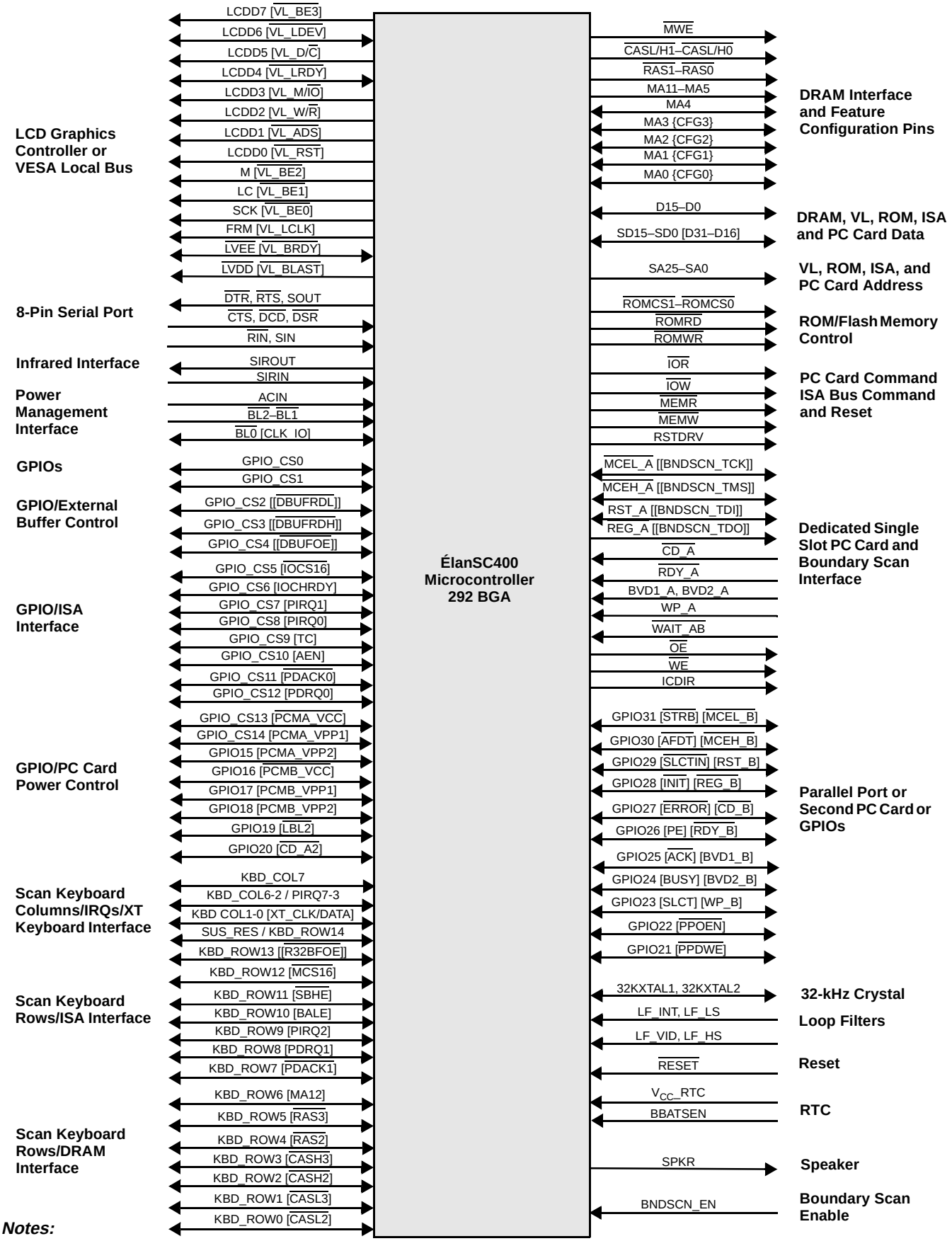
BLOCK DIAGRAM—ÉlanSC400 MICROCONTROLLER



BLOCK DIAGRAM—ÉlanSC410 MICROCONTROLLER



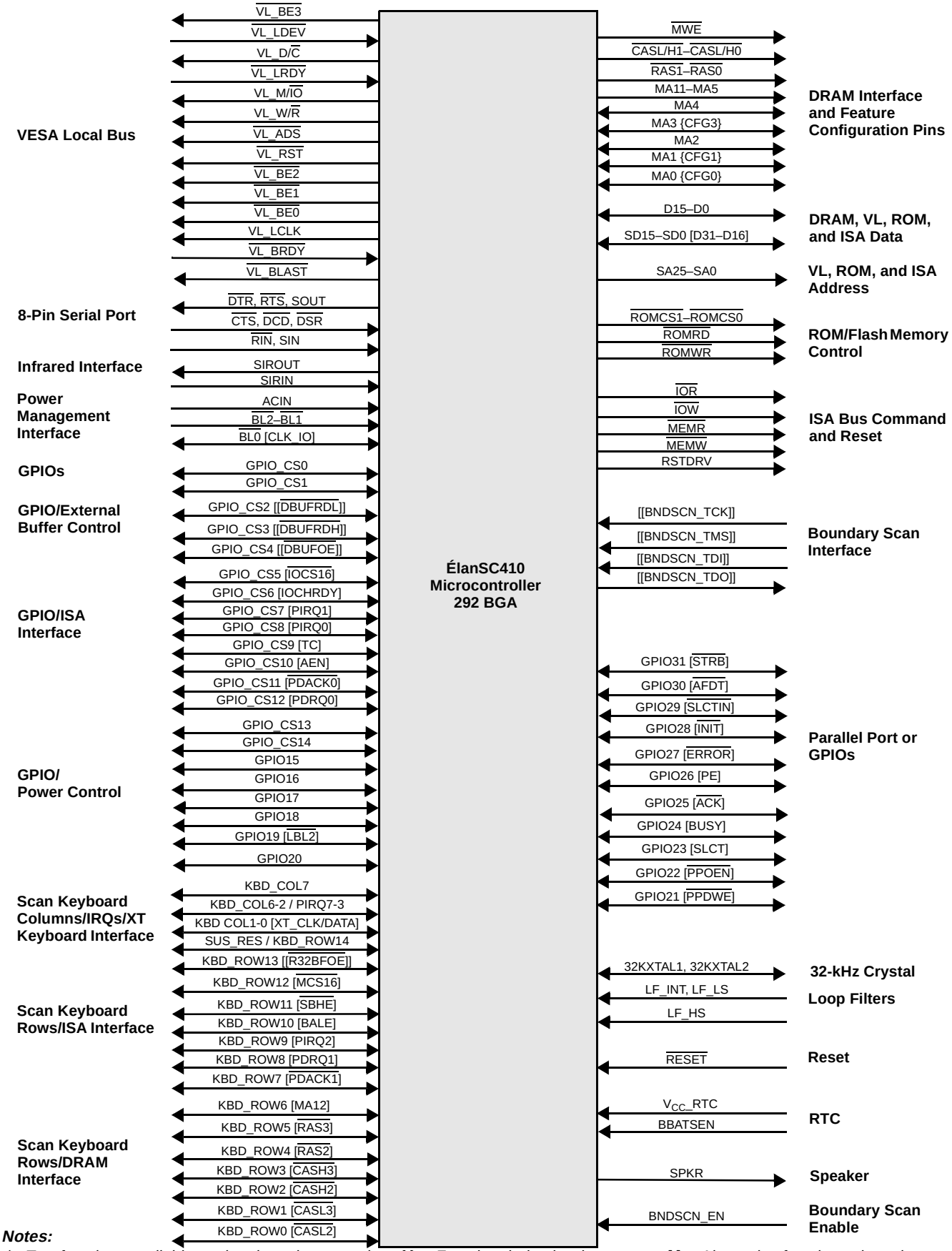
LOGIC SYMBOL—ÉlanSC400 MICROCONTROLLER



Notes:

/ =Two functions available on the pin at the same time. { } = Function during hardware reset. [ ] = Alternative function selected by firmware configuration. [[ ]] = Alternate function selected by a hardware configuration pin state at power-on reset. This does not apply to [[BNDSCN_TCK]], [[BNDSCN_TMS]], [[BNDSCN_TDI]], and [[BNDSCN_TDO]]. These alternate functions are enabled by the BNDSCN_EN signal.

LOGIC SYMBOL—ÉLANSC410 MICROCONTROLLER



Notes:

/ =Two functions available on the pin at the same time. { } = Function during hardware reset. [ ] = Alternative function selected by firmware configuration. [[ ]] = Alternate function selected by a hardware configuration pin state at power-on reset. This does not apply to [[BNDSCN_TCK]], [[BNDSCN_TMS]], [[BNDSCN_TDI]], and [[BNDSCN_TDO]]. These functions are enabled by the BNDSCN_EN signal.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

RELATED AMD PRODUCTS

E86™ Family Devices

Device	Description
80C186	16-bit microcontroller
80C188	16-bit microcontroller with 8-bit external data bus
80L186	Low-voltage, 16-bit microcontroller
80L188	Low-voltage, 16-bit microcontroller with 8-bit external data bus
Am186™EM	High-performance, 80C186-compatible, 16-bit embedded microcontroller
Am188™EM	High-performance, 80C188-compatible, 16-bit embedded microcontroller with 8-bit external data bus
Am186EMLV	High-performance, 80C186-compatible, low-voltage, 16-bit embedded microcontroller
Am188EMLV	High-performance, 80C188-compatible, low-voltage, 16-bit embedded microcontroller with 8-bit external data bus
Am186ES	High-performance, 80C186-compatible, 16-bit embedded microcontroller
Am188ES	High-performance, 80C188-compatible, 16-bit embedded microcontroller with 8-bit external data bus
Am186ESLV	High-performance, 80C186-compatible, low-voltage, 16-bit embedded microcontroller
Am188ESLV	High-performance, 80C188-compatible, low-voltage, 16-bit embedded microcontroller with 8-bit external data bus
Am186ED	High-performance, 80C186- and 80C188-compatible, 16-bit embedded microcontroller with 8- or 16-bit external data bus
Am186EDLV	High-performance, 80C186- and 80C188-compatible, low-voltage, 16-bit embedded microcontroller with 8- or 16-bit external data bus
Am186ER	High-performance, 80C186-compatible, low-voltage, 16-bit embedded microcontroller with 32 Kbyte of internal RAM
Am188ER	High-performance, 80C188-compatible, low-voltage, 16-bit embedded microcontroller with 8-bit external data bus and 32 Kbyte of internal RAM
Am186CC	High-performance, 80C186-compatible 16-bit embedded communications controller
Am186CH	High-performance, 80C186-compatible 16-bit embedded HDLC microcontroller
Am186CU	High-performance, 80C186-compatible 16-bit embedded USB microcontroller
Élan™SC300	High-performance, highly integrated, low-voltage, 32-bit embedded microcontroller
ÉlanSC310	High-performance, single-chip, 32-bit embedded PC/AT microcontroller
ÉlanSC400	Single-chip, low-power, PC/AT-compatible microcontroller
ÉlanSC410	Single-chip, PC/AT-compatible microcontroller
Am386®DX	High-performance, 32-bit embedded microprocessor with 32-bit external data bus
Am386®SX	High-performance, 32-bit embedded microprocessor with 16-bit external data bus
Am486®DX	High-performance, 32-bit embedded microprocessor with 32-bit external data bus

Related Documents

The following documents provide additional information regarding the ÉlanSC400 and ÉlanSC410 microcontrollers.

- *ÉlanSC400 and ÉlanSC410 User's Manual*, order #21030
- *ÉlanSC400 Register Set Reference Manual*, order #21032
- *ÉlanSC400 Register Set Reference Manual Amendment*, order #21032A/1
- *ÉlanSC400 Evaluation Board User's Manual*, order #21906
- *ÉlanSC400 Microcontroller and Windows CE µforCE Demonstration System User's Manual*, order #21892
- $\overline{ROMCS0}$ *Redirection to PC Card Socket A on the ÉlanSC400 Microcontroller Application Note*, order #21643

Élan™SC400 Microcontroller Evaluation Board

The Élan™SC400 microcontroller evaluation board is a stand-alone evaluation platform for the ÉlanSC400 and ÉlanSC410 microcontrollers.

As a test and development platform for designs based on the ÉlanSC400 and ÉlanSC410 microcontrollers, this AMD product is used by system designers to experiment with design trade-offs, make power measurements, and develop software. Contact your local AMD sales office for more information on evaluation board availability and pricing.

Third-Party Development Support Products

The FusionE86℠ 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 protocol stacks, emulators, hardware and software debuggers, board-level products, and software development tools, among others.

In addition, mature development tools and applications for the x86 platform are widely available in the general marketplace.

Customer Service

The AMD customer service network includes U.S. offices, international offices, and a customer training center. Expert technical assistance is available from the AMD worldwide staff of field application engineers and factory support staff to answer E86™ and Comm86™ family hardware and software development questions.

Hotline and World Wide Web Support

For answers to technical questions, AMD provides e-mail support as well as a toll-free number for direct access to our corporate applications hotline.

The AMD World Wide Web home page provides the latest product information, including technical information and data on upcoming product releases. In addition, EPD CodeKit software on the Web site provides tested source code example applications.

Corporate Applications Hotline

(800) 222-9323	Toll-free for U.S. and Canada
44-(0) 1276-803-299	U.K. and Europe hotline

Additional contact information is listed on the back of this datasheet. For technical support questions on all E86 and Comm86 products, send e-mail to **epd.support@amd.com**.

World Wide Web Home Page

To access the AMD home page, go to: **www.amd.com**. Then follow the **Embedded Processors** link for information about E86 and Comm86 products.

Questions, requests, and input concerning AMD's WWW pages can be sent via e-mail to **webmaster@amd.com**.

Documentation and Literature

Free information such as data books, user's manuals, data sheets, application notes, the *E86™ Family Products and Development Tools CD*, order #21058, and other literature is available with a simple phone call. Internationally, contact your local AMD sales office for product literature. Additional contact information is listed on the back of this data sheet.

Literature Ordering

(800) 222-9323	Toll-free for U.S. and Canada
(512) 602-5651	Direct dial worldwide
(512) 602-7639	Fax

ARCHITECTURAL OVERVIEW

The architectural goals of the ÉlanSC400 and ÉlanSC410 microcontrollers included a focus on CPU performance, CPU-to-memory performance, and internal graphics controller (ÉlanSC400 microcontroller only) performance. The resulting architecture includes several distinguishing features of interest to the system designer:

- The main system DRAM is shared between the CPU and graphics controller, so that the graphics controller can be serviced quickly to maintain video display performance at higher panel resolutions. The internal unified memory architecture (UMA) implemented on the ÉlanSC400 and ÉlanSC410 microcontrollers means lower cost and less complication for the system designer, with only one DRAM interface, fewer pins, and a much smaller board for many designs.
- CPU-to-memory performance is critical for both DRAM and ROM accesses. The CPU on the ÉlanSC400 microcontroller has a concurrent path to the ROM/Flash memory interface and can execute code out of ROM/Flash memory at the same time as the graphics controller is accessing DRAM for a screen refresh. Many system designs can take advantage of this concurrency without sacrificing performance.

- The ROM/Flash memory interface provides the flexibility to optimize the performance of ROM cycles, including the support of burst-mode ROMs. This is beneficial because products based on the ÉlanSC400 and ÉlanSC410 microcontrollers may be implemented such that the operating system or application programs are executed from ROM.
- Because the microcontrollers support a large number of external buses and interfaces, the address and data buses are shared between the various interfaces to reduce pin count on the chip.

These features result in a versatile architecture that can use various combinations of data bus sizes to achieve cost and performance goals. The architecture provides maximum performance and flexibility for high-end vertical applications, but contains functionality for a wider horizontal market that may demand less performance.

- A typical lower performance/lower cost system might implement 16-bit DRAM banks, an 8-bit ISA bus, an 8/16-bit PC Card bus, and use the internal graphic controller.
- A higher performance, full-featured system might include 32-bit DRAM, VL-bus to an external graphics controller, and a 16-bit ISA/PC Card bus.

The following basic data bus configuration rules apply. (A complete list of feature trade-offs to be considered in system design can be found in “System Considerations” on page 20.)

- When the internal graphics controller on the ÉlanSC400 microcontroller is enabled, DRAM is always 16 bits wide, and no 32-bit targets are supported. This is because the graphics controller needs a guaranteed short latency for adequate video performance. If either 32-bit DRAMs, 32-bit ROMs, or the VL-bus is enabled, the internal graphics controller is unavailable.

Note that, as a derivative of the original ÉlanSC400 microcontroller, the ÉlanSC410 microcontroller shares the primary architectural characteristics of the ÉlanSC400 microcontroller described above, minus the graphics controller and PCMCIA interfaces.

The following sections provide an overview of the features of the ÉlanSC400 and ÉlanSC410 microcontrollers, including on-chip peripherals and system interfaces.

Low-Voltage Am486 CPU Core

The ÉlanSC400 and ÉlanSC410 microcontrollers are based on the low-voltage Am486 CPU core. The core includes the following features:

- 2.7–3.3-V operation reduces power consumption
- Industry-standard 8-Kbyte unified code and data write-back cache improves both CPU and total system performance by significantly reducing traffic on the DRAM bus.
- System management mode (SMM) facilitates designs requiring power management by providing a mechanism to control power to unneeded peripherals transparently to application software.

To reduce power consumption, the floating-point unit has been removed from the Am486 CPU core. Floating-point instructions are not supported on the ÉlanSC400 and ÉlanSC410 microcontrollers, although normal software emulation can be easily implemented.

The ÉlanSC400 and ÉlanSC410 microcontrollers use the industry-standard 486 instruction set. Software written for the 486 microprocessor and previous members of the x86 architecture family can run on the ÉlanSC400 and ÉlanSC410 microcontrollers.

Power Management

Power management on the ÉlanSC400 and ÉlanSC410 microcontrollers includes a dedicated power management unit and additional power management features built into each integrated peripheral. The ÉlanSC400 and ÉlanSC410 microcontrollers can use the following techniques to conserve power:

- Slow down clocks when the system is not in active use
- Shut off clocks to parts of the chip that are idle
- Switch off power to parts of the system that are idle
- Automatically reduce power use when batteries are low

The power management unit (PMU) controls stopping and changing clocks, SMI generation, timers, activities, and battery-level monitoring. It provides:

- Hyper-Speed, High-Speed, Low-Speed, Temporary Low-Speed, Standby, Suspend, and Critical Suspend modes
- Dynamically adjusted clock speeds for power reduction
- Programmable activity and wake-up monitoring
- General-purpose I/O signals to control external devices and external power management
- Battery low and AC power monitoring
- SMI/NMI synchronization and generation

Clock Generation

The ÉlanSC400 and ÉlanSC410 microcontrollers require only one 32.768-kHz crystal to generate all the other clock frequencies required by the system. The output of the on-chip crystal oscillator circuit is used to generate the various frequencies by utilizing four Phase-Locked Loop (PLL) circuits (three for the ÉlanSC410 microcontroller). An additional PLL in the CPU is used for Hyper-Speed mode.

ROM/Flash Memory Interface

The integrated ROM/Flash memory interface supports the following features:

- 8-, 16-, and 32-bit ROM/Flash memory interfaces
- Three ROM/Flash memory chip selects
- Burst-mode ROMs
- ROM accesses at both ISA and CPU speeds (normal and fast-speed modes)
- Dedicated ROM Read and ROM Write signals for better performance

Each ROM space can accommodate up to 64 Mbyte of ROM. The three ROM spaces can be individually write-protected. This is useful for protecting code residing in Flash memory devices.

Two of the three ROM/Flash memory chip selects can be remapped to a PC Card socket via pinstrap or software control. This feature supports reprogramming of soldered-down Flash memory boot devices and also simplifies testing of BIOS/XIP OS code.

Three ROM access modes are supported: Normal mode, Fast mode, and Burst mode. A different set of timings is used in each mode. In Normal ROM access mode, the bus cycles follow ISA-like timings. In Fast ROM access mode, the bus cycle timing occurs at the CPU clock rate with controls for wait-state insertion. Burst ROM access timing is used when the ROM/Flash memory interface is fulfilling an internal CPU burst request to support a cache line refill.

Wait states are supported for all ROM and Flash memory accesses, including Burst mode. Burst-mode (page-mode) ROM reads are supported for either a 16- or 32-bit ROM interface running in Fast mode.

DRAM Controller

The integrated DRAM controller provides the signals and associated timing necessary to support an external DRAM array with minimal software programming and overhead. Internal programmable registers are provided to select the DRAM type and operating mode, as well as refresh options. A wide variety of commodity DRAMs are supported, and substantial flexibility is built into the DRAM controller to optimize performance of the CPU and (on the ÉlanSC400 microcontroller) the internal graphics controller, which uses system DRAM for its buffers.

The DRAM controller supports the following features:

- 3.3-V, 70-ns DRAMs
- Up to four banks
- 16-bit or 32-bit banks
- Up to 64 Mbyte of total memory
- Self-refresh DRAMs
- Fast page and Extended Data Out (EDO) DRAMs
- Two-way interleaved operation among identically populated banks using fast-page mode devices
- Mixed depth and width of DRAM banks in non-interleaved mode
- Symmetrical and asymmetrical DRAM support

Integrated Standard PC/AT Peripherals

The ÉlanSC400 and ÉlanSC410 microcontrollers include all the standard peripheral controllers that make up a PC/AT system.

Dual DMA Controllers

Dual, cascaded, 8237A-compatible DMA controllers provide seven user-definable DMA channels. Of the seven internal channels, four are 8-bit channels and three are 16-bit channels. Channel 4 is used for the cascade function.

Any two of the seven channels can be mapped simultaneously to external DMA request/acknowledge lines.

The DMA controller on the ÉlanSC400 and ÉlanSC410 microcontrollers is software compatible with the PC/AT cascaded 8237 controller pair. Its features include:

- Single, block, and demand transfer modes
- Enable/disable channel controller
- Address increment or decrement
- Software priority
- 64-Mbyte system address space for increased performance
- Dynamic clock-enable design for reducing clocked elements during DMA inactivity
- Programmable clock frequency for performance

Dual Interrupt Controllers

Dual, cascaded, 8259-compatible programmable interrupt controllers support 15 user-definable interrupt levels. Eight external interrupt requests can be mapped to any of the 15 internal IRQ inputs.

The interrupt controller block includes these features:

- Software-compatibility with PC/AT interrupt controllers
- 15-level priority controller
- Programmable interrupt modes
- Individual interrupt request mask capability
- Accepts requests from peripherals
- Resolves priority on pending interrupts and interrupts in service
- Issues interrupt request to processor
- Provides interrupt vectors for interrupt service routines
- Tied into the PMU for power management

The interrupt controller block is functionally compatible with the standard cascaded 8259A controller pair as implemented in the PC/AT system. The master controller drives the CPU's interrupt input signal based on the highest priority interrupt request pending at the master controller's IRQ7–IRQ0 inputs. The master IRQ2 input is configured for Cascade mode and is driven only by the slave controller's interrupt output signal. The highest pending interrupt at the slave's IRQ inputs will therefore drive the IRQ2 input of the master.

The interrupt controller has programmable sources for interrupts that are controlled through extended configuration registers and, on the ÉlanSC400 microcontroller, through PC Card controller configuration registers.

Programmable Interval Timer (PIT)

The programmable interval timer (PIT) on the ÉlanSC400 and ÉlanSC410 microcontrollers is software-compatible with PC/AT 8254 system timers. The PIT provides three 16-bit counters that can be operated independently in six different modes. The PIT is generally used for timing external events, counting, and producing repetitive waveforms. The PIT can be programmed to count in binary or in BCD.

Real-Time Clock (RTC)

The RTC designed into the ÉlanSC400 and ÉlanSC410 microcontrollers is compatible with the MC146818A device used in PC/AT systems. The RTC consists of a time-of-day clock with alarm interrupt and a 100-year calendar. The clock/calendar has a programmable periodic interrupt, 114 bytes of static user RAM, and can be represented in either binary or BCD. The RTC includes the following features:

- Counts seconds, minutes, and hours of the day
- Counts days of the week, date, month, and year
- 12–24 hour clock with AM and PM indication in 12-hour mode
- 14 clock, status, and control registers
- 114 bytes of general-purpose RAM
- Three separately software-maskable and testable interrupts
  - Time-of-day alarm is programmable to occur from once-per-second to once-per-day
  - Periodic interrupts can be continued to occur at rates from 122 µs to 500 ms
  - Update-ended interrupt provides cycle status
- Dedicated power pin directly supports lithium backup battery when the rest of the chip is completely powered down (RTC-only mode)
- Voltage monitor circuit checks the voltage level of the lithium backup battery and sets a bit when the battery is below specification.
- Internal RTC reset signal performs a reset when power is applied to the RTC core.

PC/AT Support Features

The ÉlanSC400 and ÉlanSC410 microcontrollers provide all of the support functions found in the original IBM PC/AT. These include the Port B status and control bits, speaker control, CPU-core reset based on the system control processor (SCP), and A20 gate control, as well as extensions for fast CPU core reset. In addition, a CPU shutdown cycle (e.g., as a result of a triple fault) generates a CPU core reset.

Bidirectional Enhanced Parallel Port (EPP)

The parallel port on the ÉlanSC400 and ÉlanSC410 microcontrollers is functionally compatible with IBM PC/AT and PS/2 systems, with an added EPP mode for faster transfers. The microcontroller's parallel port interface provides all the status inputs, control outputs, and the control signals necessary for the external parallel port data buffers.

The parallel port interface on both microcontrollers is shared with some of the GPIO signals and, on the ÉlanSC400 microcontroller, with the second PC Card socket interface. Only one of these interfaces can be enabled at one time.

The parallel port interface can be configured to operate in one of three different modes of operation:

- **PC/AT Compatible mode**: This mode provides a byte-wide forward (host-to-peripheral) channel with data and status lines used according to their original (Centronics) definitions in the IBM PC/AT.
- **Bidirectional mode**: This mode offers byte-wide bidirectional parallel data transfers between host and peripheral, equivalent to the parallel interface on the IBM PS/2.
- **Enhanced Parallel Port (EPP) mode**: This mode provides a byte-wide bidirectional channel controlled by the microcontroller. It provides separate address and data cycles over the eight data lines of the interface with an automatic address and data strobe for the address and data cycles, respectively. EPP mode offers wider system bandwidth and increased performance over both the PC/AT Compatible and Bidirectional modes.

Serial Port

The ÉlanSC400 and ÉlanSC410 microcontrollers include an industry-standard 16550A UART. The UART can be used to drive a standard 8-pin serial interface or a 2-pin infrared interface. The serial interface and infrared interface signals are available on the ÉlanSC400 and ÉlanSC410 microcontrollers at all times, though only one is available at any given time.

The UART powers up as a 16450-compatible device. It can be switched to and from the FIFO (16550) mode under software control. In the FIFO mode, the receive and the transmit circuitry are each enhanced by separate 16-byte FIFOs to off-load the CPU from repetitive service routines.

The serial port includes the following features:

- Eight-pin interface: serial in, serial out, two modem control lines, and four modem status lines
- Separately enabled receiver line status, receiver data, character timeout, transmitter holding register, and modem status interrupts
- Baud-rate generator provides input clock divisor from 1 to 65535 to create 16x clock
- 5-, 6-, 7-, or 8-bit data
- Even, odd, stick, or no parity generation and checking
- 1, 1-1/2 or 2 stop-bit generation
- Break generation/detection

Keyboard Interfaces

The integrated keyboard controller has the following features:

- Matrix keyboard support with up to 15 rows and 8 columns
- Hardware support for software emulation of the System Control Processor (SCP) emulation logic
- XT keyboard interface

Programmable General-Purpose Inputs and Outputs

The chip supports several general-purpose I/O signals (GPIOs) that can be used on the system board. There are two classifications of GPIO available: the GPIOx signals, which are programmable as inputs or outputs only, and the GPIO_CSx signals.

The GPIO_CSx signals have many programmable options. They can be configured as chip selects. As outputs, these signals are individually programmable to be High or Low for the following PMU modes: Hyper, High-Speed, Low-Speed, Standby, and Suspend. As inputs, they can be programmed to cause System Management Interrupts (SMIs), Non-Maskable Interrupts (NMIs), wake-ups, or activities for the power management unit. They can also be used as I/O or memory chip selects.

Infrared Port for Wireless Communication

The ÉlanSC400 and ÉlanSC410 microcontrollers support infrared data transfer. This support consists of adding additional transmit and receive serializers as well as a controlling state machine and DMA interface to the internal UART.

The integrated infrared port includes these features:

- Low-speed mode supports all bit rates from UART, up to 115 Kbit/s
- High-speed mode transfers 1.152 Mbit/s using DMA

Dual PC Card Controller (ÉlanSC400 Microcontroller Only)

The PC Card host bus adapter included on the ÉlanSC400 microcontroller conforms to *PCMCIA Standard Release 2.1*. It provides support for two sockets, each implementing the PC Card memory and I/O interfaces. The PC Card controller is not supported on the ÉlanSC410 microcontroller.

The PC Card controller includes the following features:

- ExCA-compliant, 82365-register-set compatible
- 8-bit and 16-bit data bus
- DMA transfers between I/O PC cards and system DRAM
- Ten available memory windows, five per socket

Of the two PC Card sockets supported, only one is available in all modes of operation. The second socket is multiplexed with the parallel port and GPIO features.

Register set compatibility with the 82365SL PC Card Interface Controller is maintained where features are common to both controllers.

Of the ten memory windows available, six are dedicated to the PC Card controller and four are shared with memory mapping system (MMS) Windows C–F.

Two of the three ROM/Flash memory chip selects can be remapped to a PC Card socket via pinstrap or software control. This feature supports reprogramming of soldered down Flash memory boot devices and also simplifies testing of BIOS/XIP OS code.

Graphics Controller for CGA-Compatible Text and Graphics (ÉlanSC400 Microcontroller Only)

The graphics controller included on the ÉlanSC400 microcontroller offers a low-cost integrated graphics solution for the mobile terminal market. Integration with the main processor and system logic affords the advan-

tages of an integrated local-bus interface and frame and font buffers that are shared with main memory. The graphics controller is not supported on the ÉlanSC410 microcontroller.

The graphics controller includes the following features:

- Supports multiple panel resolutions
- Provides internal unified memory architecture (UMA) with optional write-through caching of graphics buffers
- Stores frame and font buffer data in system DRAM, eliminates extra memory chip
- Provides software compatibility with Color Graphics Adapter (CGA), Monochrome Display Adapter (MDA), and Hercules Graphics Adapter (HGA) text and graphics
- Supports single-scan or dual-scan monochrome LCD panels with 4-bit or 8-bit data interface
- Typical panels supported include:
  - 640 x 200, 640 x 240, 640 x 480, 480 x 320, 480 x 240, 480 x 128, 320 x 200, 320 x 240
  - Other resolutions can be supported
- Supports single-scan color STN panels with 8-bit interface, same resolutions as monochrome mode
- Internal local-bus interface provides high performance
- Logical screen can be larger than physical window.
- Supports panning and scrolling
- Supports horizontal dot doubling and vertical line doubling

The following MDA/CGA-compatible text mode features are supported:

- 40, 64, or 80 columns with characters 16, 10, or 8 pixels wide
- Variable height characters up to 32 lines
- Variable width characters—8, 10, or 16 pixels
- MDA Monochrome, or CGA 4 gray shades, 16 gray shades, or 16-colors
- 16-Kbyte downloadable font area, relocatable on 16-Kbyte boundaries within lower 16 Mbytes of system DRAM (can be write protected)
- 16-Kbyte frame buffer, relocatable on either 16-Kbyte boundaries within lower 16 Mbyte of system DRAM (CGA-compatible mode) or 32-Kbyte boundaries when the frame buffer is larger than 16 Kbyte (flat-mapped mode)

The following graphics mode features are supported:

- 640 x 200 1 bit-per-pixel, CGA-compatible graphics buffer memory map
- 320 x 200 2 bits-per-pixel, CGA-compatible graphics buffer memory map
- 640 x 480 2 bits-per-pixel, flat memory map (lower resolutions supported)
- 640 x 480 1 bit-per-pixel, flat memory map
- 1, 2, or 4 bits-per-pixel packed-pixel flat-mapped graphics up to 640 x 240/480 x 320 with two mapping modes:
  - 16-Kbyte window with bank swapping to address up to 64 Kbyte of graphics frame buffer while consuming only 16 Kbyte of DOS/Real-mode CPU address space
  - Direct-mapped (no bank swapping) with locatable base address, up to 128-Kbyte direct addressability
- Hercules Graphics mode emulation (HGA)

JTAG Test Features

The ÉlanSC400 and ÉlanSC410 microcontrollers provide a boundary-scan interface based on the *IEEE Std 1149.1, Standard Test Access Port and Boundary-Scan Architecture*. The test access port provides a scan interface for testing the microcontroller and system hardware in a production environment. It contains extensions that allow a hardware-development system to control and observe the microcontroller without interposing hardware between the microcontroller and the system.

System Interfaces

Data Buses

The ÉlanSC400 and ÉlanSC410 microcontrollers provide 32 bits of data that are divided into two separate 16-bit buses.

- **System Data Bus**: The system (or peripheral) data bus (SD15–SD0) is always 16 bits wide and is shared between ISA, 8-bit or 16-bit ROM/Flash memory, and PC Card peripherals (ÉlanSC400 microcontroller only). It can be directly connected to all of these devices. In addition, these signals are the upper word of the VESA local (VL) data bus, the 32-bit DRAM interface, and the 32-bit ROM interface.
- **Data Bus**: The D15–D0 data bus is used during 16-bit DRAM cycles. For 32-bit DRAM, VL-bus, and ROM cycles, this bus is combined with the system data bus. In other words, the data bus signals (D31–D16) are shared with the system data bus signals SD15–SD0.

The ÉlanSC400 and ÉlanSC410 microcontrollers support the data bus configurations listed below. External transceivers or buffers can be used to isolate the buses.

- 16-bit DRAM bus, 8-/16-bit ROM, 32-bit VL-bus disabled, internal graphics controller enabled/disabled
- 16-/32-bit DRAM bus, 8/16-bit ROM, 32-bit VL-bus enabled/disabled, internal graphics controller disabled
- 16-/32-bit DRAM bus, 32-bit ROM, 32-bit VL-bus enabled/disabled, internal graphics controller disabled

See Figure 2 on page 22 and Figure 3 on page 23 for block diagrams of example systems.

The ÉlanSC400 and ÉlanSC410 microcontrollers offer flexibility in configuring the ROM and DRAM data buses for different widths. The widths (8/16/32 bits) for ROMCS0 are programmed during power-up through two pinstraps, CFG0 and CFG1. The DRAM widths (16/32 bits) are programmed through configuration registers. Up to four 16- or 32-bit banks of DRAM are supported.

Two of the three ROM/Flash memory chip selects (ROMCS2–ROMCS0) can be remapped to a PC Card socket via pinstrap or software control. This feature supports reprogramming of soldered-down Flash memory boot devices and also simplifies testing of BIOS/XIP (execute in place) OS code.

Address Buses

There are two external address buses on the ÉlanSC400 and ÉlanSC410 microcontrollers.

- **System Address Bus**: The SA25–SA0 system address bus outputs the physical memory or I/O port latched addresses. These addresses are used by all external peripheral devices other than main system DRAM. In addition, the system address bus is the local address bus in VL-bus mode.
- **DRAM Address Bus**: DRAM row and column addresses are multiplexed onto the DRAM address bus (MA12–MA0). Row addresses are driven onto this bus and are valid upon the falling edge of RAS. Column addresses are driven onto this bus and are valid upon the falling edge of CAS.

The SA bus is shared between the ISA bus, the VL-bus, the ROM/Flash memory controller and, on the ÉlanSC400 microcontroller, the PC Card controller. The ÉlanSC400 and ÉlanSC410 microcontrollers provide programmable drive strengths in the I/O buffers to accommodate loading for various system configurations.

Memory Management

The ÉlanSC400 and ÉlanSC410 microcontrollers manage up to nine separate physical device memory address spaces. All but the ISA memory address space can have a depth of up to 64 Mbyte each. The ISA bus memory area is limited to 16 Mbyte, as defined by ISA specifications. The microcontroller will drive all 26 address lines on ISA cycles to allow up to 64-Mbyte address space, as described in the memory management section of the *ÉlanSC400 and ÉlanSC410 Microcontrollers User's Manual* (order #21030)—refer to the subsection on ISA bus addressing). The nine memory spaces are:

- System memory address space (DRAM)
- ROM0 memory address space (ROMCS0 signal)
- ROM1 memory address space (ROMCS1 signal)
- ROM2 memory address space (ROMCS2 signal)
- PC Card Socket A memory address spaces (common and attribute) (ÉlanSC400 microcontroller only)
- PC Card Socket B memory address spaces (common and attribute) (ÉlanSC400 microcontroller only)
- External ISA/VL-bus memory address space

The system memory address space (DRAM) is accessible using direct-mapped CPU addresses and can also be accessed by the CPU in an indirect method using the Memory Mapping System (MMS). On the ÉlanSC400 microcontroller, DRAM is also accessible by the integrated graphics controller if enabled.

The ROM0 address space is partially accessible via a direct mapping of the CPU address bus and partially accessible via the MMS. The ROM1 and ROM2 address spaces are only accessible indirectly using the MMS.

On the ÉlanSC400 microcontroller, the PC Card address spaces are accessed through a separate, 82365SL-compatible address mapping system.

The ISA/VL-bus address space is accessible as a direct mapping of the CPU address bus. ISA memory cycles are generated when the CPU generates a memory cycle that is not detected as an access to any other memory space. An ISA bus memory cycle can also be generated if the CPU generates a memory address that resides in the ISA overlapping memory region window. This window can be defined to overlay any system memory region below 16 Mbyte.

ISA Bus Interface For External ISA Peripherals

The ISA interface consists of a subset of ISA-compatible bus signals, allowing for the connection of 8- or 16-bit devices supporting ISA-compatible I/O, memory, and DMA cycles. The following features are supported:

- 8.2944-MHz maximum bus clock speed
- Programmable DMA clock speed up to 16 MHz
- 8-bit and 16-bit ISA I/O and memory cycles (ISA memory is non-cacheable)
- Direct connection to 3- or 5-volt peripherals

Eight programmable IRQ input signals are available. These interrupts can be routed via software to any available PC/AT-compatible interrupt channel.

Two programmable DMA channels are available for external DMA peripherals. These DMA channels can be routed to software to any available ISA DMA channel.

VESA Local (VL) Bus Interface Supports 32-Bit Memory and I/O Targets

The VESA local (VL) bus controller provides the signals and associated timing necessary to support a single VESA compliant VL-bus target. Multiple VL-bus targets can be supported using external circuitry to allow multiple VL devices to share the VL_LDEV signal. This allows the ÉlanSC400 and ÉlanSC410 microcontrollers to operate as a normal VL-bus motherboard controller, in accordance with the *VL-Bus Standard 2.0*.

On the ÉlanSC400 microcontroller, the VL-bus is available only when the internal graphics controller is disabled.

The microcontroller's VL-bus controller includes the following features:

- 33-MHz operation at 3.3 V
- 32-bit data bus
- Burst-mode transfers
- Register control of local bus reset

VESA bus mastering and DMA transfers to and from the VL-bus target are not supported. VL memory is non-cacheable.

SYSTEM CONSIDERATIONS

Figure 1 shows the ÉlanSC400 microcontroller as it might be used in a minimal system design.

Figure 2 and Figure 3 show more complex system designs for each microcontroller and the features that are traded for others because of pin multiplexing.

- The ÉlanSC400 and ÉlanSC410 microcontrollers support a maximum of 4 banks of 32-bit DRAM, but because the $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ signals for the high word and for banks 2 and 3 are traded for keyboard row signals, the minimum system would have one or two banks of DRAM (either Bank 0 or Bank 1) populated with 16-bit DRAMs. The MA12 signal for asymmetrical support is also traded with a keyboard row signal.
- Because the VL-bus and the graphics controller share control signals on the ÉlanSC400 microcontroller, use of the internal graphics controller is traded with having an external VL-bus on that microcontroller.
- If either 32-bit DRAMs, 32-bit ROMs, or the VL-bus is enabled, the internal graphics controller on the ÉlanSC400 microcontroller is unavailable because of internal design constraints.
- The ÉlanSC400 and ÉlanSC410 microcontrollers provide an absolute minimum of dedicated ISA control signals. Any additional ISA controls are traded with GPIOs or keyboard rows and columns.
- The SD buffer shares control signals with some of the GPIOs. This buffer controls the high word of the D data bus (D31–D16). Note that using the SD buffer is optional. The high word of the D data bus can be hooked up directly to devices that want the SD data bus (SD15–SD0). Buffering aids in voltage translation or isolation for heavy loading.
- The $\overline{\text{R32BFOE}}$ signal buffers the high word of the D data bus (D31–D16) for 32-bit ROMs. The control signal associated with the ROM32 buffer is shared with a keyboard row.
- On the ÉlanSC400 microcontroller, the parallel port is traded for PC Card Socket B. It requires an external buffer and latch.
- The serial and infrared ports share the same internal UART. Real-time switching between the two is supported; however, only one port is available at any given time.
- $\overline{\text{ROMCS2}}$ is not connected to a dedicated pin. Software can enable and map it to any of the 15 GPIO_CS signals.

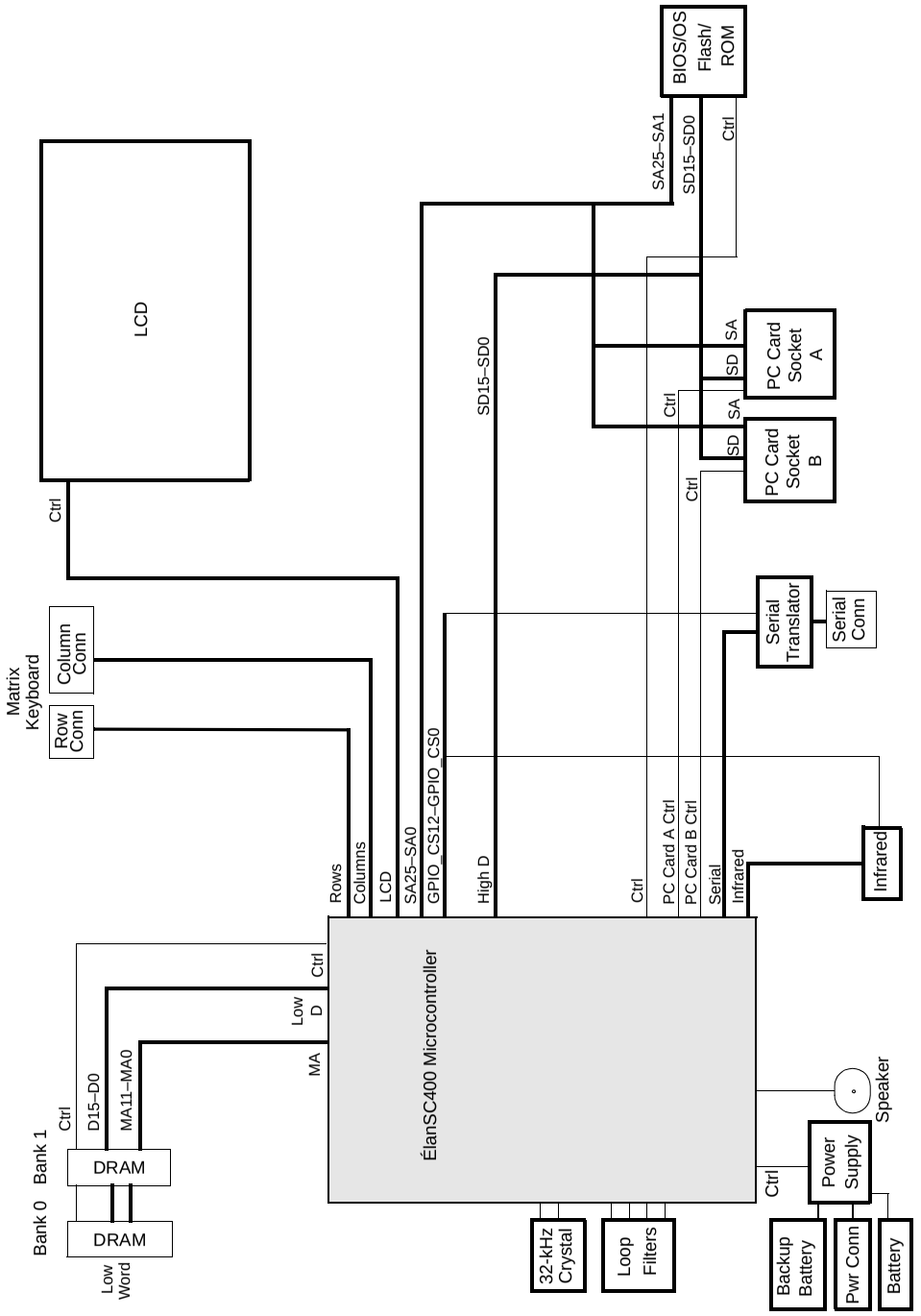
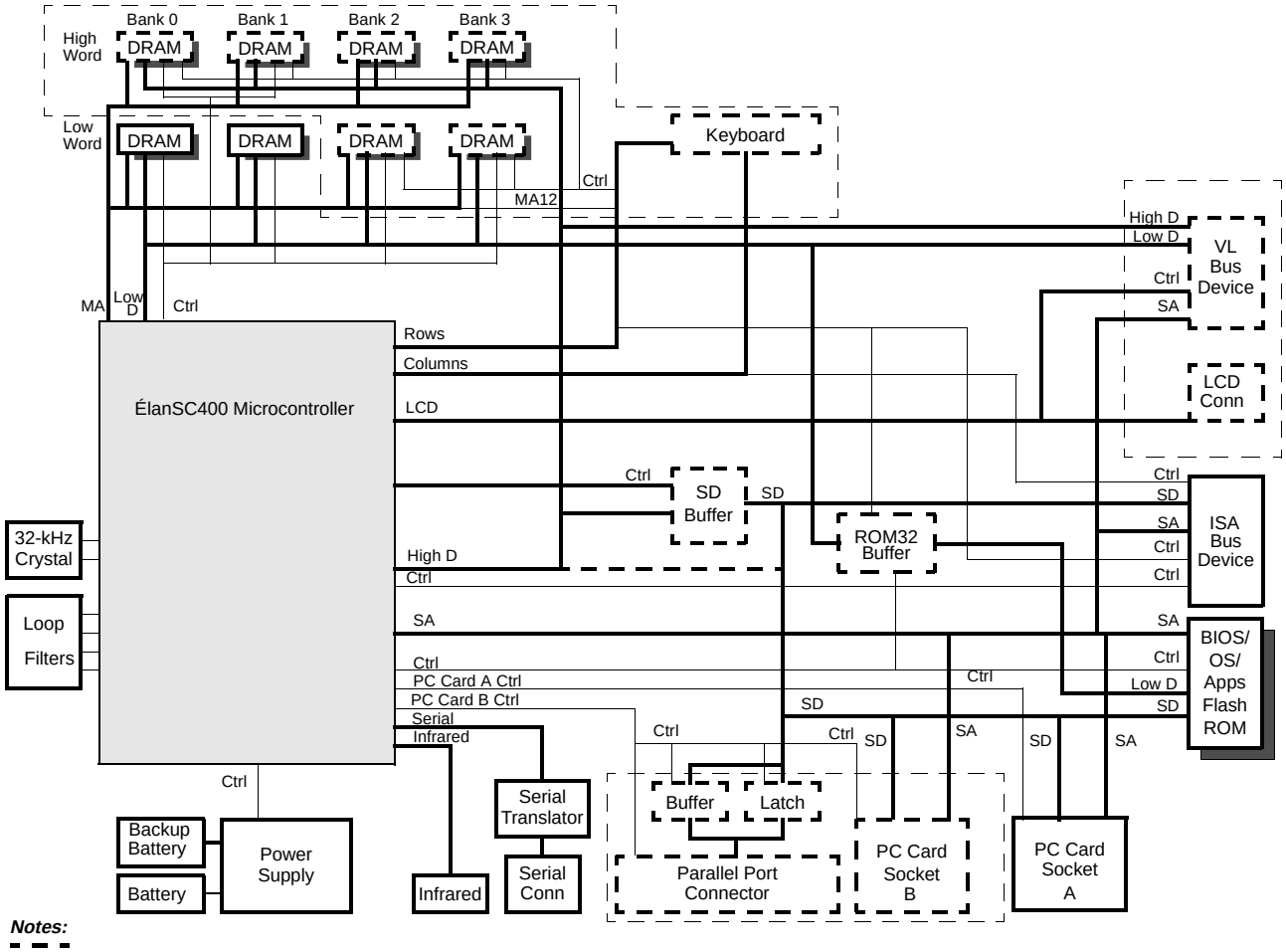


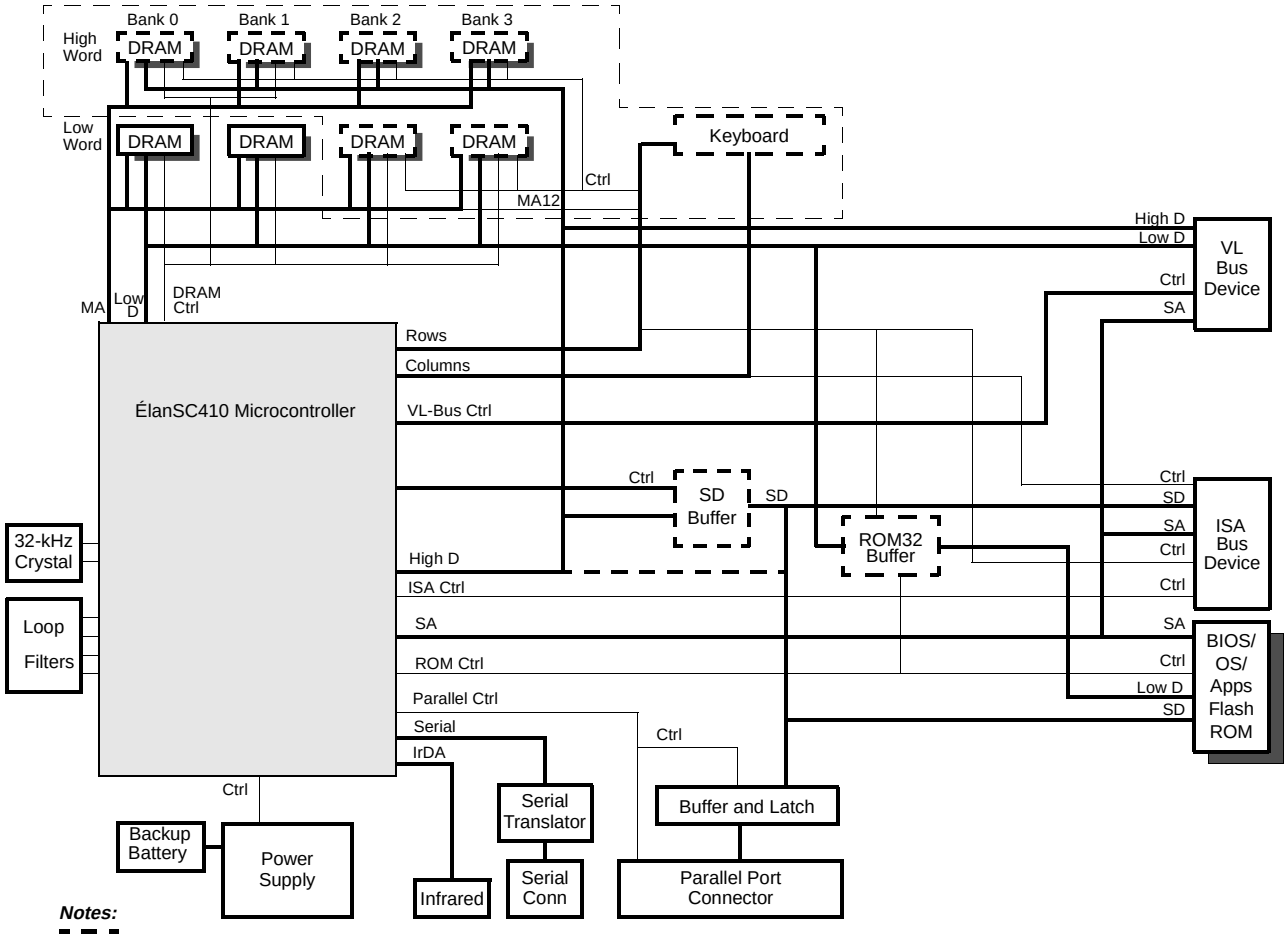
Figure 1. Typical Mobile Terminal Design



Notes:

A dashed box indicates a feature that is optional or is traded for another.

Figure 2. System Diagram with Trade-offs—ÉlanSC400 Microcontroller



Notes:

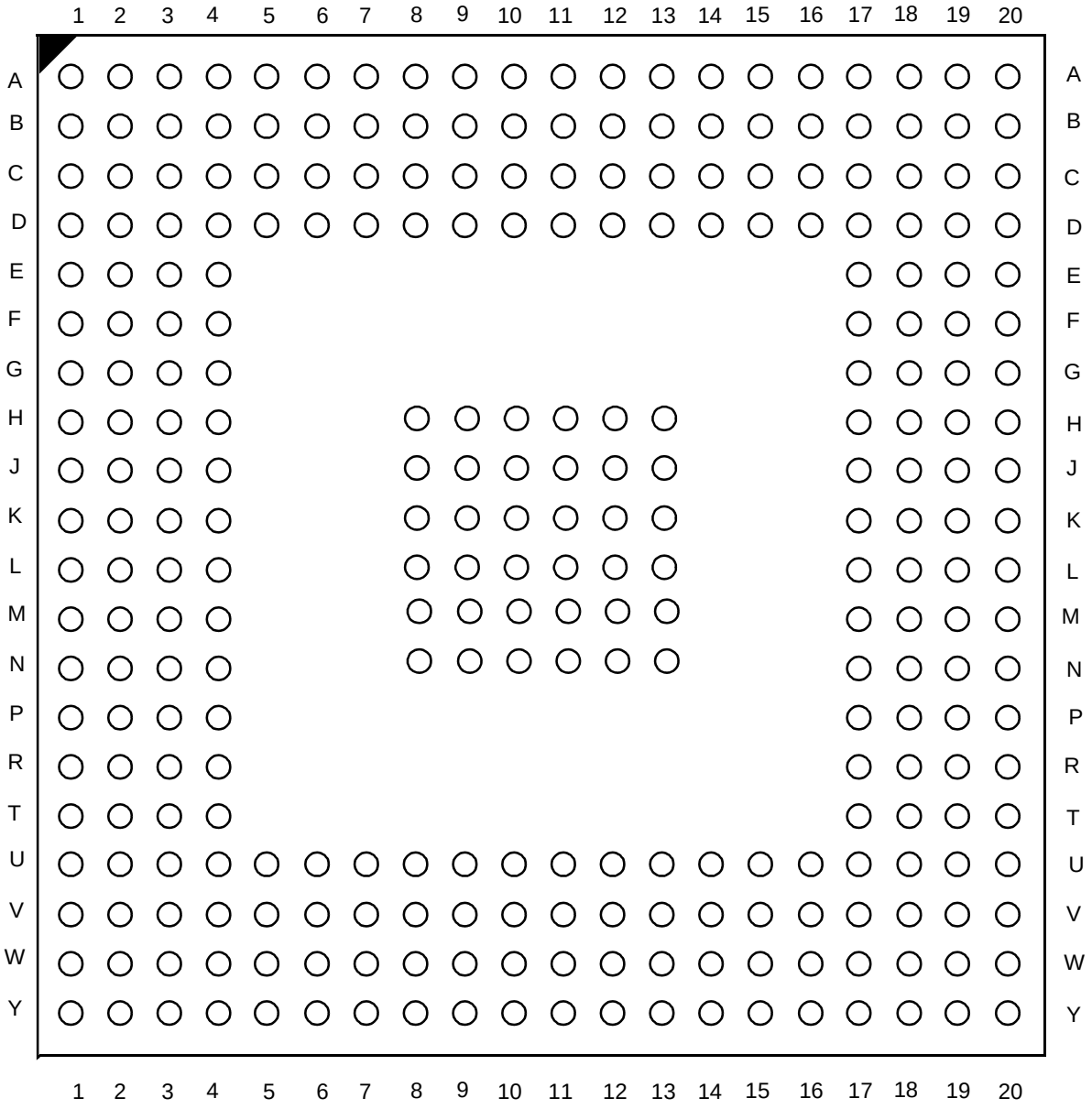
A dashed box indicates a feature that is optional or is traded for another.

Figure 3. System Design with Trade-offs—ÉlanSC410 Microcontroller

CONNECTION DIAGRAM—ÉlanSC400 AND ÉlanSC410 MICROCONTROLLERS

292 Ball Grid Array (BGA) Package

Top View (from component side looking through to bottom)



PIN DESIGNATIONS

This section identifies the pins of the ÉlanSC400 and ÉlanSC410 microcontrollers and lists the signals associated with each pin.

Several different tables are included in this section.

- The Pin Designations (Pin Number)—ÉlanSC400 Microcontroller table beginning on page 26 lists the ÉlanSC400 microcontroller signals sorted by pin number.

  The Pin Designations (Pin Number)—ÉlanSC410 Microcontroller table on page 33 lists the ÉlanSC410 microcontroller signals sorted by pin number.

  Along with the Connection Diagram on page 24, these tables can be used to associate the complete pin name (including all multiplexed functions) with the physical pin on the BGA package.
- The Pin Designations (Pin Name)—ÉlanSC400 Microcontroller table on page 29 lists the ÉlanSC400 microcontroller signals sorted in alphabetical order.

  The Pin Designations (Pin Name)—ÉlanSC410 Microcontroller table on page 36 lists the ÉlanSC410 microcontroller signals sorted in alphabetical order.

  All multiplexed signals are included in these lists. Note that these tables should not be used to determine primary and secondary functions for multiplexed pins because the ordering was changed to alphabetize every function. Please refer to the Pin Designations (Pin Number) table or the Pin State tables for the definitive listing of primary and secondary functions in the correct order for each pin.
- The Pin State tables beginning on page 42, which group pins alphabetically by function, show pin states during reset, normal operation, and Suspend mode, along with output drive strength, maximum load, supply source, and power-down groups.
- The Signal Description table beginning on page 62 includes complete pin descriptions in alphabetical order by function.
- The table beginning on page 70 clarifies the configuration options for those pins having multiple functions.

Pin Naming

The **Signal Name** column in the Pin Designation tables beginning on page 26 and in the Pin State tables beginning on page 40 is decoded as follows:

NAME1/NAME2 {NAME3} [NAME4] [[NAME5]]

NAME1

This is the only function for the pin.

NAME1/NAME2

The slash separates two functions that are available on the pin at the same time (i.e., at different times in the same design the pin is used for different functions).

{NAME3}

The name in braces is the pin function during a hardware reset.

[NAME4]

The name in square brackets is the alternative function for the pin, selected by firmware configuration. Only one function is available for each configuration.

[[NAME5]]

The name inside double square brackets is the alternate function for the pin, selected by a hardware configuration pin state at power-on reset. This does not apply to [[BNDSCN_TCK]], [[BNDSCN_TMS]], [[BNDSCN_TDI]], and [[BNDSCN_TDO]]. These four alternate functions are enabled by the BNDSCN_EN signal. Only one function is available for each configuration.

PIN CHANGES FOR THE ÉlanSC410 MICROCONTROLLER

The following signals supported on the ÉlanSC400 microcontroller are not available on the ÉlanSC410 microcontroller.

- Configuration signal: CFG2
- PC Card controller signals: $\overline{\text{MCEL\_A}}$, $\overline{\text{MCEL\_B}}$, $\overline{\text{MCEH\_A}}$, $\overline{\text{MCEH\_B}}$, $\overline{\text{RST\_A}}$, $\overline{\text{RST\_B}}$, $\overline{\text{REG\_A}}$, $\overline{\text{REG\_B}}$, $\overline{\text{CD\_A}}$, $\overline{\text{CD\_B}}$, $\overline{\text{CD\_A2}}$, RDY_A, RDY_B, $\overline{\text{BVD1\_A}}$, $\overline{\text{BVD1\_B}}$, $\overline{\text{BVD2\_A}}$, $\overline{\text{BVD2\_B}}$, $\overline{\text{WP\_A}}$, WP_B, $\overline{\text{WAIT\_AB}}$, $\overline{\text{OE}}$, $\overline{\text{WE}}$, ICDIR, $\overline{\text{PCMA\_VCC}}$, PCMA_VPP1, PCMA_VPP2, PCMB_VCC, PCMB_VPP1, PCMB_VPP2
- Graphics controller signals: LCDD7–LCDD0, M, LC, SCK, FRM, $\overline{\text{LVEE}}$, $\overline{\text{LVDD}}$
- Loop filter signal: LF_VID

PIN DESIGNATIONS (Pin Number)—ÉlanSC400 Microcontroller

Pin No.	Signal Name	Pin No.	Signal Name	Pin No.	Signal Name
A1	KBD_COL5/PIRQ6	**B19**	KBD_ROW0 [CASL2]	**D17**	GPIO_CS3 [[DBUFRDH]]
A2	KBD_COL2/PIRQ3	**B20**	LCDD0 [VL_RST]	**D18**	LCDD2 [VL_W/R]
A3	KBD_ROW13 [[R32BFOE]]	**C1**	KBD_ROW11 [SBHE]	**D19**	LCDD4 [VL_LRDY]
A4	D15	**C2**	KBD_ROW8 [PDRQ1]	**D20**	LCDD7 [VL_BE3]
A5	D12	**C3**	KBD_COL4/PIRQ5	**E1**	V_{CC}
A6	D9	**C4**	GPIO_CS4 [[DBUFOE]]	**E2**	KBD_COL0 [XT_DATA]
A7	D7	**C5**	KBD_COL7	**E3**	KBD_ROW9 [PIRQ2]
A8	V_{CC}	**C6**	D13	**E4**	GND
A9	D4	**C7**	D10	**E17**	V_{CC}
A10	D1	**C8**	D6	**E18**	LCDD5 [VL_D/C]
A11	MWE	**C9**	D2	**E19**	FRM [VL_LCLK]
A12	MA2 {CFG2}	**C10**	MA0 {CFG0}	**E20**	LC [VL_BE1]
A13	V_{CC}	**C11**	MA4	**F1**	SD4 [D20]
A14	MA5	**C12**	MA7	**F2**	SD1 [D17]
A15	MA8	**C13**	MA10	**F3**	KBD_ROW12 [MCS16]
A16	MA11	**C14**	CASL1	**F4**	GND
A17	CASH1	**C15**	RAS0	**F17**	LCDD6 [VL_LDEV]
A18	V_{CC}	**C16**	KBD_ROW5 [RAS3]	**F18**	M [VL_BE2]
A19	LVDD [VL_BLAST]	**C17**	KBD_ROW2 [CASH2]	**F19**	SCK [VL_BE0]
A20	LVEE [VL_BRDY]	**C18**	GPIO_CS2 [[DBUFRDL]]	**F20**	SA24
B1	V_{CC}	**C19**	LCDD1 [VL_ADS]	**G1**	SD6 [D22]
B2	KBD_COL6/PIRQ7	**C20**	LCDD3 [VL_M/IO]	**G2**	SD3 [D19]
B3	KBD_COL3/PIRQ4	**D1**	KBD_COL1 [XT_CLK]	**G3**	SD0 [D16]
B4	V_{CC}	**D2**	KBD_ROW10 [BALE]	**G4**	GND
B5	D14	**D3**	KBD_ROW7 [PDACK1]	**G17**	V_{CC}
B6	D11	**D4**	GND	**G18**	GPIO20 [CD_A2]
B7	D8	**D5**	GND	**G19**	SA22
B8	D5	**D6**	GND	**G20**	SA21
B9	D3	**D7**	GND	**H1**	V_{CC}
B10	D0	**D8**	GND	**H2**	SD5 [D21]
B11	MA1 {CFG1}	**D9**	GND	**H3**	SD2 [D18]
B12	MA3 {CFG3}	**D10**	GND	**H4**	GND
B13	MA6	**D11**	GND	**H8**	GND
B14	MA9	**D12**	GND	**H9**	GND
B15	CASL0	**D13**	CASH0	**H10**	GND
B16	V_{CC}	**D14**	RAS1	**H11**	GND
B17	KBD_ROW6 [MA12]	**D15**	KBD_ROW4 [RAS2]	**H12**	GND
B18	KBD_ROW3 [CASH3]	**D16**	KBD_ROW1 [CASL3]	**H13**	GND

PIN DESIGNATIONS (Pin Number)—ÉlanSC400 Microcontroller (Continued)

Pin No.	Signal Name	Pin No.	Signal Name	Pin No.	Signal Name
H17	SA25	**L10**	GND	**P3**	$\overline{\text{RDY\_A}}$
H18	SA23	**L11**	GND	**P4**	V_{CC}_CPU
H19	SA20	**L12**	GND	**P17**	V_{CC}
H20	SA18	**L13**	GND	**P18**	V_{CC}
J1	SD10 [D26]	**L17**	V_{CC}	**P19**	SA1
J2	SD7 [D23]	**L18**	SA10	**P20**	SA4
J3	V_{CC}	**L19**	SA9	**R1**	RST_A [[BNDSCN_TDI]]
J4	GND	**L20**	SA11	**R2**	$\overline{\text{CD\_A}}$
J8	GND	**M1**	V_{CC}	**R3**	BVD2_A
J9	GND	**M2**	$\overline{\text{REG\_A}}$ [[BDNSCN_TDO]]	**R4**	V_{CC}_CPU
J10	GND	**M3**	ICDIR	**R17**	GND
J11	GND	**M4**	V_{CC}_CPU	**R18**	$\overline{\text{ROMCS0}}$
J12	GND	**M8**	GND	**R19**	$\overline{\text{IOW}}$
J13	GND	**M9**	GND	**R20**	SA2
J17	V_{CC}	**M10**	GND	**T1**	V_{CC}
J18	SA19	**M11**	GND	**T2**	WP_A
J19	SA17	**M12**	GND	**T3**	GPIO22 [$\overline{\text{PPOEN}}$]
J20	SA14	**M13**	GND	**T4**	V_{CC}_CPU
K1	SD11 [D27]	**M17**	V_{CC}	**T17**	GND
K2	SD9 [D25]	**M18**	SA7	**T18**	$\overline{\text{MEMW}}$
K3	SD8 [D24]	**M19**	V_{CC}	**T19**	$\overline{\text{ROMCS1}}$
K4	V_{CC}_CPU	**M20**	SA8	**T20**	SA0
K8	GND	**N1**	SD14 [D30]	**U1**	$\overline{\text{WAIT\_AB}}$
K9	GND	**N2**	$\overline{\text{WE}}$	**U2**	GPIO25 [$\overline{\text{ACK}}$] [BVD1_B]
K10	GND	**N3**	$\overline{\text{MCEH\_A}}$ [[BNDSCN_TMS]]	**U3**	GPIO24 [BUSY] [BVD2_B]
K11	GND	**N4**	V_{CC}_CPU	**U4**	GPIO23 [SLCT] [WP_B]
K12	GND	**N8**	GND	**U5**	GND
K13	GND	**N9**	GND	**U6**	GND
K17	SA16	**N10**	GND	**U7**	GND
K18	SA15	**N11**	GND	**U8**	GND
K19	SA13	**N12**	GND	**U9**	GND
K20	SA12	**N13**	GND	**U10**	GND
L1	SD12 [D28]	**N17**	V_{CC}	**U11**	GND
L2	SD13 [D29]	**N18**	SA3	**U12**	GND
L3	SD15 [D31]	**N19**	SA5	**U13**	GND
L4	V_{CC}_CPU	**N20**	SA6	**U14**	GND
L8	GND	**P1**	$\overline{\text{OE}}$	**U15**	GND
L9	GND	**P2**	$\overline{\text{MCEL\_A}}$ [[BNDSCN_TCK]]	**U16**	GND

PIN DESIGNATIONS (Pin Number)—ÉlanSC400 Microcontroller (Continued)

Pin No.	Signal Name	Pin No.	Signal Name	Pin No.	Signal Name
U17	GND	**V19**	GPIO_CS0	**Y1**	V_{CC}
U18	V_{CC}	**V20**	$\overline{\text{ROMRD}}$	**Y2**	GPIO28 [$\overline{\text{INIT}}$] [$\overline{\text{REG\_B}}$]
U19	$\overline{\text{ROMWR}}$	**W1**	GPIO30 [$\overline{\text{AFDT}}$] [$\overline{\text{MCEH\_B}}$]	**Y3**	LF_INT
U20	$\overline{\text{IOR}}$	**W2**	GPIO26 [PE] [$\overline{\text{RDY\_B}}$]	**Y4**	32KXTAL2
V1	BVD1_A	**W3**	GPIO29 [$\overline{\text{SLCTIN}}$] [RST_B]	**Y5**	GND_ANALOG
V2	GPIO31 [$\overline{\text{STRB}}$] [$\overline{\text{MCEL\_B}}$]	**W4**	LF_LS	**Y6**	32KXTAL1
V3	GPIO21 [$\overline{\text{PPDWE}}$]	**W5**	LF_VID	**Y7**	$\overline{\text{RESET}}$
V4	GPIO27 [$\overline{\text{ERROR}}$] [$\overline{\text{CD\_B}}$]	**W6**	V_{CC}_A	**Y8**	$\overline{\text{DTR}}$
V5	LF_HS	**W7**	V_{CC}_RTC	**Y9**	SIRIN
V6	BBATSEN	**W8**	$\overline{\text{RTS}}$	**Y10**	SOUT
V7	SPKR	**W9**	V_{CC}	**Y11**	BNDSCN_EN
V8	SIROUT	**W10**	$\overline{\text{DSR}}$	**Y12**	SUS_RES/KBD_ROW14
V9	$\overline{\text{DCD}}$	**W11**	SIN	**Y13**	$\overline{\text{BL1}}$
V10	$\overline{\text{CTS}}$	**W12**	ACIN	**Y14**	GPIO18 [PCMB_VPP2]
V11	$\overline{\text{RIN}}$	**W13**	$\overline{\text{BL2}}$	**Y15**	GPIO15 [PCMA_VPP2]
V12	RSTDRV	**W14**	$\overline{\text{BL0}}$ [CLK_IO]	**Y16**	V_{CC}
V13	V_{CC}	**W15**	GPIO17 [PCMB_VPP1]	**Y17**	GPIO_CS12 [PDRQ0]
V14	GPIO19 [$\overline{\text{LBL2}}$]	**W16**	GPIO_CS14 [PCMA_VPP1]	**Y18**	GPIO_CS9 [TC]
V15	GPIO16 [$\overline{\text{PCMB\_VCC}}$]	**W17**	GPIO_CS11 [$\overline{\text{PDACK0}}$]	**Y19**	GPIO_CS7 [PIRQ1]
V16	GPIO_CS13 [$\overline{\text{PCMA\_VCC}}$]	**W18**	GPIO_CS8 [PIRQ0]	**Y20**	GPIO_CS1
V17	GPIO_CS10 [AEN]	**W19**	GPIO_CS5 [$\overline{\text{IOCS16}}$]		
V18	GPIO_CS6 [IOCHRDY]	**W20**	$\overline{\text{MEMR}}$		

PIN DESIGNATIONS (Pin Name)—ÉlanSC400 Microcontroller

Signal Name	Pin No.	Signal Name	Pin No.	Signal Name	Pin No.
ACIN	W12	**D2**	C9	**GND**	D4
[ACK] [BVD1_B] GPIO25	U2	**D3**	B9	**GND**	D5
[AEN] GPIO_CS10	V17	**D4**	A9	**GND**	D6
[AFDT] [MCEH_B] GPIO30	W1	**D5**	B8	**GND**	D7
[BALE] KBD_ROW10	D2	**D6**	C8	**GND**	D8
BBATSEN	V6	**D7**	A7	**GND**	D9
BL0 [CLK_IO]	W14	**D8**	B7	**GND**	D10
BL1	Y13	**D9**	A6	**GND**	D11
BL2	W13	**D10**	C7	**GND**	D12
BNDSCN_EN	Y11	**D11**	B6	**GND**	E4
[[BNDSCN_TCK]] MCEL_A	P2	**D12**	A5	**GND**	F4
[[BNDSCN_TDI]] RST_A	R1	**D13**	C6	**GND**	G4
[[BNDSCN_TDO]] REG_A	M2	**D14**	B5	**GND**	H4
[[BNDSCN_TMS]] MCEH_A	N3	**D15**	A4	**GND**	H8
[BUSY] [BVD2_B] GPIO24	U3	**[D16] SD0**	G3	**GND**	H9
BVD1_A	V1	**[D17] SD1**	F2	**GND**	H10
[BVD1_B] GPIO25 [ACK]	U2	**[D18] SD2**	H3	**GND**	H11
BVD2_A	R3	**[D19] SD3**	G2	**GND**	H12
[BVD2_B] GPIO24 [BUSY]	U3	**[D20] SD4**	F1	**GND**	H13
CASH0	D13	**[D21] SD5**	H2	**GND**	J4
CASH1	A17	**[D22] SD6**	G1	**GND**	J8
[CASH2] KBD_ROW2	C17	**[D23] SD7**	J2	**GND**	J9
[CASH3] KBD_ROW3	B18	**[D24] SD8**	K3	**GND**	J10
CASL0	B15	**[D25] SD9**	K2	**GND**	J11
CASL1	C14	**[D26] SD10**	J1	**GND**	J12
[CASL2] KBD_ROW0	B19	**[D27] SD11**	K1	**GND**	J13
[CASL3] KBD_ROW1	D16	**[D28] SD12**	L1	**GND**	K8
CD_A	R2	**[D29] SD13**	L2	**GND**	K9
[CD_A2] GPIO20	G18	**[D30] SD14**	N1	**GND**	K10
[CD_B] GPIO27 [ERROR]	V4	**[D31] SD15**	L3	**GND**	K11
{CFG0} MA0	C10	**[[DBUFOE]] GPIO_CS4**	C4	**GND**	K12
{CFG1} MA1	B11	**[[DBUFRDH]] GPIO_CS3**	D17	**GND**	K13
{CFG2} MA2	A12	**[[DBUFRDL]] GPIO_CS2**	C18	**GND**	L8
{CFG3} MA3	B12	**DCD**	V9	**GND**	L9
[CLK_IO] BL0	W14	**DSR**	W10	**GND**	L10
CTS	V10	**DTR**	Y8	**GND**	L11
D0	B10	**[ERROR] [CD_B] GPIO27**	V4	**GND**	L12
D1	A10	**FRM [VL_LCLK]**	E19	**GND**	L13

PIN DESIGNATIONS (Pin Name)—ÉlanSC400 Microcontroller (Continued)

Signal Name	Pin No.	Signal Name	Pin No.	Signal Name	Pin No.
GND	M8	GPIO_CS10 [AEN]	V17	KBD_ROW0 [CASL2]	B19
GND	M9	GPIO_CS11 [PDACK0]	W17	KBD_ROW1 [CASL3]	D16
GND	M10	GPIO_CS12 [PDRQ0]	Y17	KBD_ROW2 [CASH2]	C17
GND	M11	GPIO_CS13 [PCMA_VCC]	V16	KBD_ROW3 [CASH3]	B18
GND	M12	GPIO_CS14 [PCMA_VPP1]	W16	KBD_ROW4 [RAS2]	D15
GND	M13	GPIO15 [PCMA_VPP2]	Y15	KBD_ROW5 [RAS3]	C16
GND	N8	GPIO16 [PCMB_VCC]	V15	KBD_ROW6 [MA12]	B17
GND	N9	GPIO17 [PCMB_VPP1]	W15	KBD_ROW7 [PDACK1]	D3
GND	N10	GPIO18 [PCMB_VPP2]	Y14	KBD_ROW8 [PDRQ1]	C2
GND	N11	GPIO19 [LBL2]	V14	KBD_ROW9 [PIRQ2]	E3
GND	N12	GPIO20 [CD_A2]	G18	KBD_ROW10 [BALE]	D2
GND	N13	GPIO21 [PPDWE]	V3	KBD_ROW11 [SBHE]	C1
GND	R17	GPIO22 [PPOEN]	T3	KBD_ROW12 [MCS16]	F3
GND	T17	GPIO23 [SLCT] [WP_B]	U4	KBD_ROW13 [[R32BFOE]]	A3
GND	U5	GPIO24 [BUSY] [BVD2_B]	U3	KBD_ROW14 / SUS_RES	Y12
GND	U6	GPIO25 [ACK] [BVD1_B]	U2	[LBL2] GPIO19	V14
GND	U7	GPIO26 [PE] [RDY_B]	W2	LC [VL_BE1]	E20
GND	U8	GPIO27 [ERROR] [CD_B]	V4	LCDD0 [VL_RST]	B20
GND	U9	GPIO28 [INIT] [REG_B]	Y2	LCDD1 [VL_ADS]	C19
GND	U10	GPIO29 [SLCTIN] [RST_B]	W3	LCDD2 [VL_W/R]	D18
GND	U11	GPIO30 [AFDT] [MCEH_B]	W1	LCDD3 [VL_M/IO]	C20
GND	U12	GPIO31 [STRB] [MCEL_B]	V2	LCDD4 [VL_LRDY]	D19
GND	U13	ICDIR	M3	LCDD5 [VL_D/C]	E18
GND	U14	[INIT] [REG_B] GPIO28	Y2	LCDD6 [VL_LDEV]	F17
GND	U15	[IOCHRDY] GPIO_CS6	V18	LCDD7 [VL_BE3]	D20
GND	U16	[IOCS16] GPIO_CS5	W19	LF_HS	V5
GND	U17	IOR	U20	LF_INT	Y3
GND_ANALOG	Y5	IOW	R19	LF_LS	W4
GPIO_CS0	V19	32KXTAL1	Y6	LF_VID	W5
GPIO_CS1	Y20	32KXTAL2	Y4	LVDD [VL_BLAST]	A19
GPIO_CS2 [[DBUFRDL]]	C18	KBD_COL0 [XT_DATA]	E2	LVEE [VL_BRDY]	A20
GPIO_CS3 [[DBUFRDH]]	D17	KBD_COL1 [XT_CLK]	D1	M [VL_BE2]	F18
GPIO_CS4 [[DBUFOE]]	C4	KBD_COL2/PIRQ3	A2	MA0 {CFG0}	C10
GPIO_CS5 [IOCS16]	W19	KBD_COL3/PIRQ4	B3	MA1 {CFG1}	B11
GPIO_CS6 [IOCHRDY]	V18	KBD_COL4/PIRQ5	C3	MA2 {CFG2}	A12
GPIO_CS7 [PIRQ1]	Y19	KBD_COL5/PIRQ6	A1	MA3 {CFG3}	B12
GPIO_CS8 [PIRQ0]	W18	KBD_COL6/PIRQ7	B2	MA4	C11
GPIO_CS9 [TC]	Y18	KBD_COL7	C5	MA5	A14

PIN DESIGNATIONS (Pin Name)—ÉlanSC400 Microcontroller (Continued)

Signal Name	Pin No.	Signal Name	Pin No.	Signal Name	Pin No.
MA6	B13	**RAS0**	C15	**SA20**	H19
MA7	C12	**RAS1**	D14	**SA21**	G20
MA8	A15	**[RAS2] KBD_ROW4**	D15	**SA22**	G19
MA9	B14	**[RAS3] KBD_ROW5**	C16	**SA23**	H18
MA10	C13	**RDY_A**	P3	**SA24**	F20
MA11	A16	**[RDY_B] GPIO26 [PE]**	W2	**SA25**	H17
[MA12] KBD_ROW6	B17	**REG_A [[BNDSCN_TDO]]**	M2	**[SBHE] KBD_ROW11**	C1
MCEH_A [[BNDSCN_TMS]]	N3	**[REG_B] GPIO28 [INIT]**	Y2	**SCK [VL_BE0]**	F19
[MCEH_B] GPIO30 [AFDT]	W1	**RESET**	Y7	**SD0 [D16]**	G3
MCEL_A [[BNDSCN_TCK]]	P2	**RIN**	V11	**SD1 [D17]**	F2
[MCEL_B] GPIO31 [STRB]	V2	**ROMCS0**	R18	**SD2 [D18]**	H3
[MCS16] KBD_ROW12	F3	**ROMCS1**	T19	**SD3 [D19]**	G2
MEMR	W20	**ROMRD**	V20	**SD4 [D20]**	F1
MEMW	T18	**ROMWR**	U19	**SD5 [D21]**	H2
MWE	A11	**RST_A [[BNDSCN_TDI]]**	R1	**SD6 [D22]**	G1
OE	P1	**[RST_B] GPIO29 [SLCTIN]**	W3	**SD7 [D23]**	J2
[PCMA_VCC] GPIO_CS13	V16	**RSTDRV**	V12	**SD8 [D24]**	K3
[PCMA_VPP1] GPIO_CS14	W16	**RTS**	W8	**SD9 [D25]**	K2
[PCMA_VPP2] GPIO15	Y15	**SA0**	T20	**SD10 [D26]**	J1
[PCMB_VCC] GPIO16	V15	**SA1**	P19	**SD11 [D27]**	K1
[PCMB_VPP1] GPIO17	W15	**SA2**	R20	**SD12 [D28]**	L1
[PCMB_VPP2] GPIO18	Y14	**SA3**	N18	**SD13 [D29]**	L2
[PDACK0] GPIO_CS11	W17	**SA4**	P20	**SD14 [D30]**	N1
[PDACK1] KBD_ROW7	D3	**SA5**	N19	**SD15 [D31]**	L3
[PDRQ0] GPIO_CS12	Y17	**SA6**	N20	**SIN**	W11
[PDRQ1] KBD_ROW8	C2	**SA7**	M18	**SIRIN**	Y9
[PE] [RDY_B] GPIO26	W2	**SA8**	M20	**SIROUT**	V8
[PIRQ0] GPIO_CS8	W18	**SA9**	L19	**[SLCT] [WP_B] GPIO23**	U4
[PIRQ1] GPIO_CS7	Y19	**SA10**	L18	**[SLCTIN] [RST_B] GPIO29**	W3
[PIRQ2] KBD_ROW9	E3	**SA11**	L20	**SOUT**	Y10
PIRQ3/KBD_COL2	A2	**SA12**	K20	**SPKR**	V7
PIRQ4/KBD_COL3	B3	**SA13**	K19	**[STRB] [MCEL_B] GPIO31**	V2
PIRQ5/KBD_COL4	C3	**SA14**	J20	**SUS_RES/KBD_ROW14**	Y12
PIRQ6/KBD_COL5	A1	**SA15**	K18	**[TC] GPIO_CS9**	Y18
PIRQ7/KBD_COL6	B2	**SA16**	K17	**V_{CC}**	A8
[PPDWE] GPIO21	V3	**SA17**	J19	**V_{CC}**	A13
[PPOEN] GPIO22	T3	**SA18**	H20	**V_{CC}**	A18
[[R32BFOE]] KBD_ROW13	A3	**SA19**	J18	**V_{CC}**	B1

PIN DESIGNATIONS (Pin Name)—ÉlanSC400 Microcontroller (Continued)

Signal Name	Pin No.	Signal Name	Pin No.	Signal Name	Pin No.
V_{CC}	B4	V_{CC}	V13	[VL_BE3] LCDD7	D20
V_{CC}	B16	V_{CC}	W9	[VL_BLAST] LVDD	A19
V_{CC}	E1	V_{CC}	Y1	[VL_BRDY] LVEE	A20
V_{CC}	E17	V_{CC}	Y16	[VL_D/C] LCDD5	E18
V_{CC}	G17	V_{CC}_A	W6	[VL_LCLK] FRM	E19
V_{CC}	H1	V_{CC}_CPU	K4	[VL_LDEV] LCDD6	F17
V_{CC}	J3	V_{CC}_CPU	L4	[VL_LRDY] LCDD4	D19
V_{CC}	J17	V_{CC}_CPU	M4	[VL_M/IO] LCDD3	C20
V_{CC}	L17	V_{CC}_CPU	N4	[VL_RST] LCDD0	B20
V_{CC}	M1	V_{CC}_CPU	P4	[VL_W/R] LCDD2	D18
V_{CC}	M17	V_{CC}_CPU	R4	WAIT_AB	U1
V_{CC}	M19	V_{CC}_CPU	T4	WE	N2
V_{CC}	N17	V_{CC}_RTC	W7	WP_A	T2
V_{CC}	P17	[VL_ADS] LCDD1	C19	[WP_B] GPIO23 [SLCT]	U4
V_{CC}	P18	[VL_BE0] SCK	F19	[XT_CLK] KBD_COL1	D1
V_{CC}	T1	[VL_BE1] LC	E20	[XT_DATA] KBD_COL0	E2
V_{CC}	U18	[VL_BE2] M	F18		

PIN DESIGNATIONS (Pin Number)—ÉlanSC410 MICROCONTROLLER

Pin No.	Signal Name	Pin No.	Signal Name	Pin No.	Signal Name
A1	KBD_COL5/PIRQ6	**B19**	KBD_ROW0 [$\overline{\text{CASL2}}$]	**D17**	GPIO_CS3 [[$\overline{\text{DBUFRDH}}$]]
A2	KBD_COL2/PIRQ3	**B20**	$\overline{\text{VL\_RST}}$	**D18**	VL_W/$\overline{\text{R}}$
A3	KBD_ROW13 [[$\overline{\text{R32BFOE}}$]]	**C1**	KBD_ROW11 [$\overline{\text{SBHE}}$]	**D19**	$\overline{\text{VL\_LRDY}}$
A4	D15	**C2**	KBD_ROW8 [PDRQ1]	**D20**	$\overline{\text{VL\_BE3}}$
A5	D12	**C3**	KBD_COL4/PIRQ5	**E1**	V_{CC}
A6	D9	**C4**	GPIO_CS4 [[$\overline{\text{DBUFOE}}$]]	**E2**	KBD_COL0 [XT_DATA]
A7	D7	**C5**	KBD_COL7	**E3**	KBD_ROW9 [PIRQ2]
A8	V_{CC}	**C6**	D13	**E4**	GND
A9	D4	**C7**	D10	**E17**	V_{CC}
A10	D1	**C8**	D6	**E18**	VL_D/$\overline{\text{C}}$
A11	$\overline{\text{MWE}}$	**C9**	D2	**E19**	VL_LCLK
A12	MA2	**C10**	MA0 {CFG0}	**E20**	$\overline{\text{VL\_BE1}}$
A13	V_{CC}	**C11**	MA4	**F1**	SD4 [D20]
A14	MA5	**C12**	MA7	**F2**	SD1 [D17]
A15	MA8	**C13**	MA10	**F3**	KBD_ROW12 [$\overline{\text{MCS16}}$]
A16	MA11	**C14**	$\overline{\text{CASL1}}$	**F4**	GND
A17	$\overline{\text{CASH1}}$	**C15**	$\overline{\text{RAS0}}$	**F17**	$\overline{\text{VL\_LDEV}}$
A18	V_{CC}	**C16**	KBD_ROW5 [$\overline{\text{RAS3}}$]	**F18**	$\overline{\text{VL\_BE2}}$
A19	$\overline{\text{VL\_BLAST}}$	**C17**	KBD_ROW2 [$\overline{\text{CASH2}}$]	**F19**	$\overline{\text{VL\_BE0}}$
A20	$\overline{\text{VL\_BRDY}}$	**C18**	GPIO_CS2 [[$\overline{\text{DBUFRDL}}$]]	**F20**	SA24
B1	V_{CC}	**C19**	$\overline{\text{VL\_ADS}}$	**G1**	SD6 [D22]
B2	KBD_COL6/PIRQ7	**C20**	VL_M/$\overline{\text{IO}}$	**G2**	SD3 [D19]
B3	KBD_COL3/PIRQ4	**D1**	KBD_COL1 [XT_CLK]	**G3**	SD0 [D16]
B4	V_{CC}	**D2**	KBD_ROW10 [BALE]	**G4**	GND
B5	D14	**D3**	KBD_ROW7 [$\overline{\text{PDACK1}}$]	**G17**	V_{CC}
B6	D11	**D4**	GND	**G18**	GPIO20
B7	D8	**D5**	GND	**G19**	SA22
B8	D5	**D6**	GND	**G20**	SA21
B9	D3	**D7**	GND	**H1**	V_{CC}
B10	D0	**D8**	GND	**H2**	SD5 [D21]
B11	MA1 {CFG1}	**D9**	GND	**H3**	SD2 [D18]
B12	MA3 {CFG3}	**D10**	GND	**H4**	GND
B13	MA6	**D11**	GND	**H8**	GND
B14	MA9	**D12**	GND	**H9**	GND
B15	$\overline{\text{CASL0}}$	**D13**	$\overline{\text{CASH0}}$	**H10**	GND
B16	V_{CC}	**D14**	$\overline{\text{RAS1}}$	**H11**	GND
B17	KBD_ROW6 [MA12]	**D15**	KBD_ROW4 [$\overline{\text{RAS2}}$]	**H12**	GND
B18	KBD_ROW3 [$\overline{\text{CASH3}}$]	**D16**	KBD_ROW1 [$\overline{\text{CASL3}}$]	**H13**	GND

PIN DESIGNATIONS (Pin Number)—ÉlanSC410 MICROCONTROLLER (Continued)

Pin No.	Signal Name	Pin No.	Signal Name	Pin No.	Signal Name
H17	SA25	**L10**	GND	**P3**	Reserved
H18	SA23	**L11**	GND	**P4**	V_{CC}_CPU
H19	SA20	**L12**	GND	**P17**	V_{CC}
H20	SA18	**L13**	GND	**P18**	V_{CC}
J1	SD10 [D26]	**L17**	V_{CC}	**P19**	SA1
J2	SD7 [D23]	**L18**	SA10	**P20**	SA4
J3	V_{CC}	**L19**	SA9	**R1**	[[BNDSCN_TDI]]
J4	GND	**L20**	SA11	**R2**	Reserved
J8	GND	**M1**	V_{CC}	**R3**	Reserved
J9	GND	**M2**	[[BNDSCN_TDO]]	**R4**	V_{CC}_CPU
J10	GND	**M3**	Reserved	**R17**	GND
J11	GND	**M4**	V_{CC}_CPU	**R18**	$\overline{\text{ROMCS0}}$
J12	GND	**M8**	GND	**R19**	$\overline{\text{IOW}}$
J13	GND	**M9**	GND	**R20**	SA2
J17	V_{CC}	**M10**	GND	**T1**	V_{CC}
J18	SA19	**M11**	GND	**T2**	Reserved
J19	SA17	**M12**	GND	**T3**	GPIO22 [$\overline{\text{PPOEN}}$]
J20	SA14	**M13**	GND	**T4**	V_{CC}_CPU
K1	SD11 [D27]	**M17**	V_{CC}	**T17**	GND
K2	SD9 [D25]	**M18**	SA7	**T18**	$\overline{\text{MEMW}}$
K3	SD8 [D24]	**M19**	V_{CC}	**T19**	$\overline{\text{ROMCS1}}$
K4	V_{CC}_CPU	**M20**	SA8	**T20**	SA0
K8	GND	**N1**	SD14 [D30]	**U1**	Reserved
K9	GND	**N2**	Reserved	**U2**	GPIO25 [$\overline{\text{ACK}}$]
K10	GND	**N3**	[[BNDSCN_TMS]]	**U3**	GPIO24 [BUSY]
K11	GND	**N4**	V_{CC}_CPU	**U4**	GPIO23 [SLCT]
K12	GND	**N8**	GND	**U5**	GND
K13	GND	**N9**	GND	**U6**	GND
K17	SA16	**N10**	GND	**U7**	GND
K18	SA15	**N11**	GND	**U8**	GND
K19	SA13	**N12**	GND	**U9**	GND
K20	SA12	**N13**	GND	**U10**	GND
L1	SD12 [D28]	**N17**	V_{CC}	**U11**	GND
L2	SD13 [D29]	**N18**	SA3	**U12**	GND
L3	SD15 [D31]	**N19**	SA5	**U13**	GND
L4	V_{CC}_CPU	**N20**	SA6	**U14**	GND
L8	GND	**P1**	Reserved	**U15**	GND
L9	GND	**P2**	[[BNDSCN_TCK]]	**U16**	GND

PIN DESIGNATIONS (Pin Number)—ÉlanSC410 MICROCONTROLLER (Continued)

Pin No.	Signal Name	Pin No.	Signal Name	Pin No.	Signal Name
U17	GND	**V19**	GPIO_CS0	**Y1**	V_{CC}
U18	V_{CC}	**V20**	$\overline{\text{ROMRD}}$	**Y2**	GPIO28 [$\overline{\text{INIT}}$]
U19	$\overline{\text{ROMWR}}$	**W1**	GPIO30 [$\overline{\text{AFDT}}$]	**Y3**	LF_INT
U20	$\overline{\text{IOR}}$	**W2**	GPIO26 [PE]	**Y4**	32KXTAL2
V1	Reserved	**W3**	GPIO29 [$\overline{\text{SLCTIN}}$]	**Y5**	GND_ANALOG
V2	GPIO31 [$\overline{\text{STRB}}$]	**W4**	LF_LS	**Y6**	32KXTAL1
V3	GPIO21 [$\overline{\text{PPDWE}}$]	**W5**	Reserved	**Y7**	$\overline{\text{RESET}}$
V4	GPIO27 [$\overline{\text{ERROR}}$]	**W6**	V_{CC}_A	**Y8**	$\overline{\text{DTR}}$
V5	LF_HS	**W7**	V_{CC}_RTC	**Y9**	SIRIN
V6	BBATSEN	**W8**	$\overline{\text{RTS}}$	**Y10**	SOUT
V7	SPKR	**W9**	V_{CC}	**Y11**	BNDSCN_EN
V8	SIROUT	**W10**	$\overline{\text{DSR}}$	**Y12**	SUS_RES/KBD_ROW14
V9	$\overline{\text{DCD}}$	**W11**	SIN	**Y13**	$\overline{\text{BL1}}$
V10	$\overline{\text{CTS}}$	**W12**	ACIN	**Y14**	GPIO18
V11	$\overline{\text{RIN}}$	**W13**	$\overline{\text{BL2}}$	**Y15**	GPIO15
V12	RSTDRV	**W14**	$\overline{\text{BL0}}$ [CLK_IO]	**Y16**	V_{CC}
V13	V_{CC}	**W15**	GPIO17	**Y17**	GPIO_CS12 [PDRQ0]
V14	GPIO19 [$\overline{\text{LBL2}}$]	**W16**	GPIO_CS14	**Y18**	GPIO_CS9 [TC]
V15	GPIO16	**W17**	GPIO_CS11 [$\overline{\text{PDACK0}}$]	**Y19**	GPIO_CS7 [PIRQ1]
V16	GPIO_CS13	**W18**	GPIO_CS8 [PIRQ0]	**Y20**	GPIO_CS1
V17	GPIO_CS10 [AEN]	**W19**	GPIO_CS5 [$\overline{\text{IOCS16}}$]		
V18	GPIO_CS6 [IOCHRDY]	**W20**	$\overline{\text{MEMR}}$		

PIN DESIGNATIONS (Pin Name)—ÉlanSC410 MICROCONTROLLER

Signal Name	Pin No.	Signal Name	Pin No.	Signal Name	Pin No.
ACIN	W12	**D10**	C7	**GND**	E4
[$\overline{\text{ACK}}$] GPIO25	U2	**D11**	B6	**GND**	F4
[AEN] GPIO_CS10	V17	**D12**	A5	**GND**	G4
[$\overline{\text{AFDT}}$] GPIO30	W1	**D13**	C6	**GND**	H4
[BALE] KBD_ROW10	D2	**D14**	B5	**GND**	H8
BBATSEN	V6	**D15**	A4	**GND**	H9
$\overline{\text{BL0}}$ [CLK_IO]	W14	**[D16] SD0**	G3	**GND**	H10
$\overline{\text{BL1}}$	Y13	**[D17] SD1**	F2	**GND**	H11
$\overline{\text{BL2}}$	W13	**[D18] SD2**	H3	**GND**	H12
BNDSCN_EN	Y11	**[D19] SD3**	G2	**GND**	H13
[[BNDSCN_TCK]]	P2	**[D20] SD4**	F1	**GND**	J4
[[BNDSCN_TDI]]	R1	**[D21] SD5**	H2	**GND**	J8
[[BNDSCN_TDO]]	M2	**[D22] SD6**	G1	**GND**	J9
[[BNDSCN_TMS]]	N3	**[D23] SD7**	J2	**GND**	J10
[BUSY] GPIO24	U3	**[D24] SD8**	K3	**GND**	J11
$\overline{\text{CASH0}}$	D13	**[D25] SD9**	K2	**GND**	J12
$\overline{\text{CASH1}}$	A17	**[D26] SD10**	J1	**GND**	J13
[$\overline{\text{CASH2}}$] KBD_ROW2	C17	**[D27] SD11**	K1	**GND**	K8
[$\overline{\text{CASH3}}$] KBD_ROW3	B18	**[D28] SD12**	L1	**GND**	K9
$\overline{\text{CASL0}}$	B15	**[D29] SD13**	L2	**GND**	K10
$\overline{\text{CASL1}}$	C14	**[D30] SD14**	N1	**GND**	K11
[$\overline{\text{CASL2}}$] KBD_ROW0	B19	**[D31] SD15**	L3	**GND**	K12
[$\overline{\text{CASL3}}$] KBD_ROW1	D16	**[[$\overline{\text{DBUFOE}}$]] GPIO_CS4**	C4	**GND**	K13
{CFG0} MA0	C10	**[[$\overline{\text{DBUFRDH}}$]] GPIO_CS3**	D17	**GND**	L8
{CFG1} MA1	B11	**[[$\overline{\text{DBUFRDL}}$]] GPIO_CS2**	C18	**GND**	L9
{CFG3} MA3	B12	**$\overline{\text{DCD}}$**	V9	**GND**	L10
[CLK_IO] $\overline{\text{BL0}}$	W14	**$\overline{\text{DSR}}$**	W10	**GND**	L11
$\overline{\text{CTS}}$	V10	**$\overline{\text{DTR}}$**	Y8	**GND**	L12
D0	B10	**[$\overline{\text{ERROR}}$] GPIO27**	V4	**GND**	L13
D1	A10	**GND**	D4	**GND**	M8
D2	C9	**GND**	D5	**GND**	M9
D3	B9	**GND**	D6	**GND**	M10
D4	A9	**GND**	D7	**GND**	M11
D5	B8	**GND**	D8	**GND**	M12
D6	C8	**GND**	D9	**GND**	M13
D7	A7	**GND**	D10	**GND**	N8
D8	B7	**GND**	D11	**GND**	N9
D9	A6	**GND**	D12	**GND**	N10

PIN DESIGNATIONS (Pin Name)—ÉlanSC410 MICROCONTROLLER (Continued)

Signal Name	Pin No.	Signal Name	Pin No.	Signal Name	Pin No.
GND	N11	GPIO19 [LBL2]	V14	KBD_ROW10 [BALE]	D2
GND	N12	GPIO20	G18	KBD_ROW11 [SBHE]	C1
GND	N13	GPIO21 [PPDWE]	V3	KBD_ROW12 [MCS16]	F3
GND	R17	GPIO22 [PPOEN]	T3	KBD_ROW13 [[R32BFOE]]	A3
GND	T17	GPIO23 [SLCT]	U4	KBD_ROW14 / SUS_RES	Y12
GND	U5	GPIO24 [BUSY]	U3	[LBL2] GPIO19	V14
GND	U6	GPIO25 [ACK]	U2	LF_HS	V5
GND	U7	GPIO26 [PE]	W2	LF_INT	Y3
GND	U8	GPIO27 [ERROR]	V4	LF_LS	W4
GND	U9	GPIO28 [INIT]	Y2	MA0 {CFG0}	C10
GND	U10	GPIO29 [SLCTIN]	W3	MA1 {CFG1}	B11
GND	U11	GPIO30 [AFDT]	W1	MA2	A12
GND	U12	GPIO31 [STRB]	V2	MA3 {CFG3}	B12
GND	U13	[INIT] GPIO28	Y2	MA4	C11
GND	U14	[IOCHRDY] GPIO_CS6	V18	MA5	A14
GND	U15	[IOCS16] GPIO_CS5	W19	MA6	B13
GND	U16	IOR	U20	MA7	C12
GND	U17	IOW	R19	MA8	A15
GND_ANALOG	Y5	32KXTAL1	Y6	MA9	B14
GPIO_CS0	V19	32KXTAL2	Y4	MA10	C13
GPIO_CS1	Y20	KBD_COL0 [XT_DATA]	E2	MA11	A16
GPIO_CS2 [[DBUFRDL]]	C18	KBD_COL1 [XT_CLK]	D1	[MA12] KBD_ROW6	B17
GPIO_CS3 [[DBUFRDH]]	D17	KBD_COL2/PIRQ3	A2	[MCS16] KBD_ROW12	F3
GPIO_CS4 [[DBUFOE]]	C4	KBD_COL3/PIRQ4	B3	MEMR	W20
GPIO_CS5 [IOCS16]	W19	KBD_COL4/PIRQ5	C3	MEMW	T18
GPIO_CS6 [IOCHRDY]	V18	KBD_COL5/PIRQ6	A1	MWE	A11
GPIO_CS7 [PIRQ1]	Y19	KBD_COL6/PIRQ7	B2	[PDACK0] GPIO_CS11	W17
GPIO_CS8 [PIRQ0]	W18	KBD_COL7	C5	[PDACK1] KBD_ROW7	D3
GPIO_CS9 [TC]	Y18	KBD_ROW0 [CASL2]	B19	[PDRQ0] GPIO_CS12	Y17
GPIO_CS10 [AEN]	V17	KBD_ROW1 [CASL3]	D16	[PDRQ1] KBD_ROW8	C2
GPIO_CS11 [PDACK0]	W17	KBD_ROW2 [CASH2]	C17	[PE] GPIO26	W2
GPIO_CS12 [PDRQ0]	Y17	KBD_ROW3 [CASH3]	B18	[PIRQ0] GPIO_CS8	W18
GPIO_CS13	V16	KBD_ROW4 [RAS2]	D15	[PIRQ1] GPIO_CS7	Y19
GPIO_CS14	W16	KBD_ROW5 [RAS3]	C16	[PIRQ2] KBD_ROW9	E3
GPIO15	Y15	KBD_ROW6 [MA12]	B17	PIRQ3/KBD_COL2	A2
GPIO16	V15	KBD_ROW7 [PDACK1]	D3	PIRQ4/KBD_COL3	B3
GPIO17	W15	KBD_ROW8 [PDRQ1]	C2	PIRQ5/KBD_COL4	C3
GPIO18	Y14	KBD_ROW9 [PIRQ2]	E3	PIRQ6/KBD_COL5	A1

PIN DESIGNATIONS (Pin Name)—ÉlanSC410 MICROCONTROLLER (Continued)

Signal Name	Pin No.	Signal Name	Pin No.	Signal Name	Pin No.
PIRQ7/KBD_COL6	B2	**SA12**	K20	**[STRB] GPIO31**	V2
[PPDWE] GPIO21	V3	**SA13**	K19	**SUS_RES/KBD_ROW14**	Y12
[PPOEN] GPIO22	T3	**SA14**	J20	**[TC] GPIO_CS9**	Y18
[[R32BFOE]] KBD_ROW13	A3	**SA15**	K18	**V_{CC}**	A8
RAS0	C15	**SA16**	K17	**V_{CC}**	A13
RAS1	D14	**SA17**	J19	**V_{CC}**	A18
[RAS2] KBD_ROW4	D15	**SA18**	H20	**V_{CC}**	B1
[RAS3] KBD_ROW5	C16	**SA19**	J18	**V_{CC}**	B4
Reserved	M3	**SA20**	H19	**V_{CC}**	B16
Reserved	N2	**SA21**	G20	**V_{CC}**	E1
Reserved	P1	**SA22**	G19	**V_{CC}**	E17
Reserved	P3	**SA23**	H18	**V_{CC}**	G17
Reserved	R2	**SA24**	F20	**V_{CC}**	H1
Reserved	R3	**SA25**	H17	**V_{CC}**	J3
Reserved	T2	**[SBHE] KBD_ROW11**	C1	**V_{CC}**	J17
Reserved	U1	**SD0 [D16]**	G3	**V_{CC}**	L17
Reserved	V1	**SD1 [D17]**	F2	**V_{CC}**	M1
Reserved	W5	**SD2 [D18]**	H3	**V_{CC}**	M17
RESET	Y7	**SD3 [D19]**	G2	**V_{CC}**	M19
RIN	V11	**SD4 [D20]**	F1	**V_{CC}**	N17
ROMCS0	R18	**SD5 [D21]**	H2	**V_{CC}**	P17
ROMCS1	T19	**SD6 [D22]**	G1	**V_{CC}**	P18
ROMRD	V20	**SD7 [D23]**	J2	**V_{CC}**	T1
ROMWR	U19	**SD8 [D24]**	K3	**V_{CC}**	U18
RSTDRV	V12	**SD9 [D25]**	K2	**V_{CC}**	V13
RTS	W8	**SD10 [D26]**	J1	**V_{CC}**	W9
SA0	T20	**SD11 [D27]**	K1	**V_{CC}**	Y1
SA1	P19	**SD12 [D28]**	L1	**V_{CC}**	Y16
SA2	R20	**SD13 [D29]**	L2	**V_{CC}_A**	W6
SA3	N18	**SD14 [D30]**	N1	**V_{CC}_CPU**	K4
SA4	P20	**SD15 [D31]**	L3	**V_{CC}_CPU**	L4
SA5	N19	**SIN**	W11	**V_{CC}_CPU**	M4
SA6	N20	**SIRIN**	Y9	**V_{CC}_CPU**	N4
SA7	M18	**SIROUT**	V8	**V_{CC}_CPU**	P4
SA8	M20	**[SLCT] GPIO23**	U4	**V_{CC}_CPU**	R4
SA9	L19	**[SLCTIN] GPIO29**	W3	**V_{CC}_CPU**	T4
SA10	L18	**SOUT**	Y10	**V_{CC}_RTC**	W7
SA11	L20	**SPKR**	V7	**VL_ADS**	C19

PIN DESIGNATIONS (Pin Name)—ÉlanSC410 MICROCONTROLLER (Continued)

Signal Name	Pin No.	Signal Name	Pin No.	Signal Name	Pin No.
VL_BE0	F19	**VL_BRDY**	A20	**VL_M/IO**	C20
VL_BE1	E20	**VL_D/C**	E18	**VL_RST**	B20
VL_BE2	F18	**VL_LCLK**	E19	**VL_W/R**	D18
VL_BE3	D20	**VL_LDEV**	F17	**[XT_CLK] KBD_COL1**	D1
VL_BLAST	A19	**VL_LRDY**	D19	**[XT_DATA] KBD_COL0**	E2

PIN STATE TABLES

The pin state tables beginning on page 42 are grouped alphabetically by function and show pin states during reset, normal operation, and Suspend mode, along with output drive strength, maximum load, supply source, and power-down groups.

Pin Characteristics

The following information describes the individual column headings in the Pin State tables beginning on page 42. Most abbreviations are defined in Table 3. Drive types and power-down groups are defined in Tables 2 and 5, respectively.

- **Pin Number**: The Pin Number column in all tables identifies the pin number of the individual I/O signal on the package.
- **Type**: The abbreviations in the Type column for all tables are defined in Table 2.
- **Output Drive**: The Output Drive column designates the output drive strength of the pin. The footnote after the drive strength letter designates that the drive strength is programmable. The available drive strengths are indicated in Table 2.

Table 2. Drive Output Description

Output Drive	Ioh_{TTL}/Iol_{TTL} [1]	V_{cc}
A	-3mA/3mA	3.0 V
B	-6mA/6mA	3.0 V
C [2]	-12mA/12mA	3.0 V
D	-18mA/18mA	3.0 V
E [2]	-24mA/24mA	3.0 V

Notes:

1. The current out of a pin is given as a negative value.

2. Output drive is programmable.

- **Max Load**: The Max Load column designates the load at which the I/O timing for that pin is guaranteed. It is also used to determine derated AC timing.
- **Supply**: The Supply column identifies the V_{CC} pin that supplies power for the specified I/O pin.The pin state table shows the pin state and termination for each power management unit mode.
- **Normal Operation**: The Normal Operation column covers the following power management modes:
  - Hyper-Speed mode
  - High-Speed mode
  - Low-Speed mode
  - Temporary Low-Speed mode
  - Standby mode
- **Suspend State**: The letters used in the Suspend State column are defined in Table 3. Note that in Critical Suspend mode, pin terminations remain unchanged from the prior mode.

Table 3. Pin Type Abbreviations

Symbol	Meaning
[ ]	Brackets signify alternate state
{ }	Reset configuration pin
-	Not an output during Suspend mode
Act	Pin continues to function during Suspend mode
B	Bidirectional
H	Driven High (a logical 1)
I	Pin is an input
IOD	Input or open drain output
L	Driven Low (a logical 0)
LS	Last state of the pin prior to entering Suspend mode
NA	Not applicable
O	Pin is an active output
OD	Open drain output
OD-STI	Pin is typically an open drain output, but can be configured as a Schmitt trigger input
PD	Built-in pulldown resistor
PPD	Programmable pulldown or no resistor
PPU	Programmable pullup or no resistor
PPUD	Programmable pullup or pulldown resistor
PU	Built-in pullup resistor
S	5-V safe pin
STI	Pin is a Schmitt trigger input
STI-OD	Pin is typically a Schmitt trigger input, but can be configured as an open-drain output
TS	Three-state output

Table 4. Power Pin Type Abbreviations

Symbol	Meaning
A	Pin is an analog input
CPU	CPU power input
RTC	Real-time clock input
V_{CC}	Power input

- **Power-Down Group**: The signals on the chip are grouped together by interface for the purpose of powering down chips on the system board that are connected to these signals in Suspend mode. The letters A–I in the Power-Down Group column indicate the group with which each affected signal is associated. Only those signals that have a different Suspend state based on the interface powering off have a letter. The interfaces are identified in Table 5. The extended registers have bits that allow components connected to each interface to be powered down in Suspend mode. Care must be taken when designing a system with sections that power down, because many signals are shared between components.

Table 5. Power-Down Groups

Group	Interface
A	DRAM
B	ROM
C	ISA (shared ISA signals individually enabled)
D	Serial port, serial IrDA infrared port
E	GPIO Chip Selects 1–0
F	VL bus
G	PC Card Socket A
H	PC Card Socket B and parallel port
I	SD buffer control signals

- **5 V**: An S in the 5 V column denotes pins that are 5-volt safe. This means these signals can tolerate 5 volts and they will not be damaged. However, they cannot drive to 5 volts.

Using the Pin State Tables

In the following Pin State tables, multiplexed pins include values specific to each signal in the row (across the table) where that signal is named. If a cell has only one value listed for two or three different signals, then this value is constant (does not change) no matter what signal is programmed to come out on the pin.

- For example, in the table on page 47, when pin V14 is GPIO19, it is bidirectional; when pin V14 is $\overline{\text{LBL2}}$, it is an output only. Because the cell includes two separate lines, the pin type is unique for each signal.
- When the V14 pin is either GPIO19 or $\overline{\text{LBL2}}$, the reset state is I-PU. Because there is only one value shown in the table, this value applies to both signals.

Table 6. Pin State Table—System Interface[1]

Signal Name [Alternate Function]	Pin #	Type	Output Drive	Max Load (pF)	Supply	Reset State	Normal Operation	Suspend State	Power Down Group	Note	5 V
$\overline{\text{IOR}}$	U20	O	C	50	V_{CC}	H	O	H[TS-PD][TS]	C	2	S
$\overline{\text{IOW}}$	R19	O	C	50	V_{CC}	H	O	H[TS-PD][TS]	C	2	S
$\overline{\text{MEMR}}$	W20	O	C	50	V_{CC}	H	O	H[TS-PD][TS]	C	2	S
$\overline{\text{MEMW}}$	T18	O	C	50	V_{CC}	H	O	H[TS-PD][TS]	C	2	S
RSTDRV	V12	O	A	30	V_{CC}	H	O	TS-PD		3	
SA0	T20	O	C–E[4]	70	V_{CC}	H	O	TS-PD		3	S
SA1	P19	O	C–E[4]	70	V_{CC}	H	O	TS-PD		3	S
SA2	R20	O	C–E[4]	70	V_{CC}	H	O	TS-PD		3	S
SA3	N18	O	C–E[4]	70	V_{CC}	H	O	TS-PD		3	S
SA4	P20	O	C–E[4]	70	V_{CC}	H	O	TS-PD		3	S
SA5	N19	O	C–E[4]	70	V_{CC}	H	O	TS-PD		3	S
SA6	N20	O	C–E[4]	70	V_{CC}	H	O	TS-PD		3	S
SA7	M18	O	C–E[4]	70	V_{CC}	H	O	TS-PD		3	S
SA8	M20	O	C–E[4]	70	V_{CC}	H	O	TS-PD		3	S
SA9	L19	O	C–E[4]	70	V_{CC}	H	O	TS-PD		3	S
SA10	L18	O	C–E[4]	70	V_{CC}	H	O	TS-PD		3	S
SA11	L20	O	C–E[4]	70	V_{CC}	H	O	TS-PD		3	S
SA12	K20	O	C–E[4]	70	V_{CC}	H	O	TS-PD		3	S
SA13	K19	O	C–E[4]	70	V_{CC}	H	O	TS-PD		3	S
SA14	J20	O	C–E[4]	70	V_{CC}	H	O	TS-PD		3	S
SA15	K18	O	C–E[4]	70	V_{CC}	H	O	TS-PD		3	S
SA16	K17	O	C–E[4]	70	V_{CC}	H	O	TS-PD		3	S
SA17	J19	O	C–E[4]	70	V_{CC}	H	O	TS-PD		3	S
SA18	H20	O	C–E[4]	70	V_{CC}	H	O	TS-PD		3	S
SA19	J18	O	C–E[4]	70	V_{CC}	H	O	TS-PD		3	S
SA20	H19	O	C–E[4]	70	V_{CC}	H	O	TS-PD		3	S
SA21	G20	O	C–E[4]	70	V_{CC}	H	O	TS-PD		3	S
SA22	G19	O	C–E[4]	70	V_{CC}	H	O	TS-PD		3	S
SA23	H18	O	C–E[4]	70	V_{CC}	H	O	TS-PD		3	S
SA24	F20	O	B	50	V_{CC}	H	O	TS-PD		3	S
SA25	H17	O	B	50	V_{CC}	H	O	TS-PD		3	S
SD0 [D16]	G3	B [B]	C–E[4]	70	V_{CC}	TS-PD	B-PPUD	I-PD		5	S
SD1 [D17]	F2	B [B]	C–E[4]	70	V_{CC}	TS-PD	B-PPUD	I-PD		5	S
SD2 [D18]	H3	B [B]	C–E[4]	70	V_{CC}	TS-PD	B-PPUD	I-PD		5	S
SD3 [D19]	G2	B [B]	C–E[4]	70	V_{CC}	TS-PD	B-PPUD	I-PD		5	S
SD4 [D20]	F1	B [B]	C–E[4]	70	V_{CC}	TS-PD	B-PPUD	I-PD		5	S
SD5 [D21]	H2	B [B]	C–E[4]	70	V_{CC}	TS-PD	B-PPUD	I-PD		5	S
SD6 [D22]	G1	B [B]	C–E[4]	70	V_{CC}	TS-PD	B-PPUD	I-PD		5	S
SD7 [D23]	J2	B [B]	C–E[4]	70	V_{CC}	TS-PD	B-PPUD	I-PD		5	S
SD8 [D24]	K3	B [B]	C–E[4]	70	V_{CC}	TS-PD	B-PPUD	I-PD		5	S
SD9 [D25]	K2	B [B]	C–E[4]	70	V_{CC}	TS-PD	B-PPUD	I-PD		5	S
SD10 [D26]	J1	B [B]	C–E[4]	70	V_{CC}	TS-PD	B-PPUD	I-PD		5	S

Table 6. Pin State Table—System Interface[1] (Continued)

Signal Name [Alternate Function]	Pin #	Type	Output Drive	Max Load (pF)	Supply	Reset State	Normal Operation	Suspend State	Power Down Group	Note	5 V
SD11 [D27]	K1	B [B]	C–E[4]	70	V_{CC}	TS-PD	B-PPUD	I-PD		5	S
SD12 [D28]	L1	B [B]	C–E[4]	70	V_{CC}	TS-PD	B-PPUD	I-PD		5	S
SD13 [D29]	L2	B [B]	C–E[4]	70	V_{CC}	TS-PD	B-PPUD	I-PD		5	S
SD14 [D30]	N1	B [B]	C–E[4]	70	V_{CC}	TS-PD	B-PPUD	I-PD		5	S
SD15 [D31]	L3	B [B]	C–E[4]	70	V_{CC}	TS-PD	B-PPUD	I-PD		5	S

Notes:

1. *Pin states for AEN, IOCHRDY, $\overline{IOCS16}$, $\overline{PDACK0}$, PDRQ0, TC, and PIRQ1–PIRQ0 are listed in Table 9 on page 49. Pin states for BALE, $\overline{MCS16}$, $\overline{SBHE}$, PDACK1, PDRQ1 and PIRQ7–PIRQ2 are listed in Table 14 on page 53.*

2. *The ISA control signals have three programmable options for Suspend mode:*

   –Driven High (inactive).

   –Three-stated with no pullup or pulldown. This is useful when the ISA device is at 5 V and left powered in Suspend. The board design should not drive 3.3-V signals into a 5-V device during Suspend because this can waste power. The system designer should provide large pullup resistors to 5 V for each of these signals on the board if this configuration is programmed.

   – Three-stated with pulldown resistors when suspended with the intent of powering off the ISA device (Power-Down Group C).

   Be careful when handling $\overline{IOR}$ and $\overline{IOW}$ because they are shared with the PC Card sockets and may need to be buffered if certain combinations of system components are powered up and off.

   ***Summary:** These pins have built-in pulldown resistors that are invoked by:*

   –Suspend mode and the ISA interface is programmed to be powered off in Suspend mode (Power-Down Group C).

3. *The SA bus, SA25–SA0, and the RSTDRV signal are three-stated with pulldowns in Suspend mode. This accommodates having the ISA bus, PC Card sockets, VL bus, and ROM interfaces left powered on or powered off in Suspend mode.*

   ***Summary:** These pins have built-in pulldown resistors that are invoked by:*

   –Suspend mode.

4. C, D, and E output drives are programmable.

5. *The combination of SD15–SD0 and D31–D16 on the same pins requires the signals to be pulled up in SD bus mode (for PC compatibility) and pulled down in D bus mode (for consistency with D15–D0). Regardless of the mode the bus is in, the pins are in the input state (i.e., they are still bidirectional and are not driven as outputs) and pulled down in Suspend mode.*

   These signals are pulled up or down automatically depending on whether the SD buffer is enabled or not (CFG3), and whether the system is in Suspend mode or not.

   ***Summary:** These pins have built-in pulldown and pullup resistors that are invoked by:*

   –Reset invokes the pulldown resistors.

   –Suspend mode invokes the pulldown resistors.

   –Operating (Hyper/High/Low/Temp Low-Speed modes): the pins will have pullups if the SD buffer control signals are enabled, and have pulldowns otherwise.

Table 7. Pin State Table—Memory Interface[1]

Signal Name [Alternate Function]	Pin #	Type	Output Drive	Max Load (pF)	Supply	Reset State	Normal Operation	Suspend State	Power Down Group	Note	5 V
$\overline{\text{CASH0}}$	D13	O	D	30	V_{CC}	H	O	O[L][TS-PD]	A	2	
$\overline{\text{CASH1}}$	A17	O	D	30	V_{CC}	H	O	O[L][TS-PD]	A	2	
$\overline{\text{CASL0}}$	B15	O	D	30	V_{CC}	H	O	O[L][TS-PD]	A	2	
$\overline{\text{CASL1}}$	C14	O	D	30	V_{CC}	H	O	O[L][TS-PD]	A	2	
D0	B10	B	C–E[3]	70	V_{CC}	TS-PD	B-PD	TS-PD		4	
D1	A10	B	C–E[3]	70	V_{CC}	TS-PD	B-PD	TS-PD		4	
D2	C9	B	C–E[3]	70	V_{CC}	TS-PD	B-PD	TS-PD		4	
D3	B9	B	C–E[3]	70	V_{CC}	TS-PD	B-PD	TS-PD		4	
D4	A9	B	C–E[3]	70	V_{CC}	TS-PD	B-PD	TS-PD		4	
D5	B8	B	C–E[3]	70	V_{CC}	TS-PD	B-PD	TS-PD		4	
D6	C8	B	C–E[3]	70	V_{CC}	TS-PD	B-PD	TS-PD		4	
D7	A7	B	C–E[3]	70	V_{CC}	TS-PD	B-PD	TS-PD		4	
D8	B7	B	C–E[3]	70	V_{CC}	TS-PD	B-PD	TS-PD		4	
D9	A6	B	C–E[3]	70	V_{CC}	TS-PD	B-PD	TS-PD		4	
D10	C7	B	C–E[3]	70	V_{CC}	TS-PD	B-PD	TS-PD		4	
D11	B6	B	C–E[3]	70	V_{CC}	TS-PD	B-PD	TS-PD		4	
D12	A5	B	C–E[3]	70	V_{CC}	TS-PD	B-PD	TS-PD		4	
D13	C6	B	C–E[3]	70	V_{CC}	TS-PD	B-PD	TS-PD		4	
D14	B5	B	C–E[3]	70	V_{CC}	TS-PD	B-PD	TS-PD		4	
D15	A4	B	C–E[3]	70	V_{CC}	TS-PD	B-PD	TS-PD		4	
KBD_ROW0 [$\overline{\text{CASL2}}$]	B19	STI-OD [O]	D	250	V_{CC}	I-PU	IOD-PU O	I-PU O[L][TS-PD]	A	2	
KBD_ROW1 [$\overline{\text{CASL3}}$]	D16	STI-OD [O]	D	250	V_{CC}	I-PU	IOD-PU O	I-PU O[L][TS-PD]	A	2	
KBD_ROW2 [$\overline{\text{CASH2}}$]	C17	STI-OD [O]	D	250	V_{CC}	I-PU	IOD-PU O	I-PU O[L][TS-PD]	A	2	
KBD_ROW3 [$\overline{\text{CASH3}}$]	B18	STI-OD [O]	D	250	V_{CC}	I-PU	IOD-PU O	I-PU O[L][TS-PD]	A	2	
KBD_ROW4 [$\overline{\text{RAS2}}$]	D15	STI-OD [O]	C–E[3]	250	V_{CC}	I-PU	IOD-PU O	I-PU O[L][TS-PD]	A	2	
KBD_ROW5 [$\overline{\text{RAS3}}$]	C16	STI-OD [O]	C–E[3]	250	V_{CC}	I-PU	IOD-PU O	I-PU O[L][TS-PD]	A	2	
KBD_ROW6 [MA12]	B17	STI-OD [O]	C–E[3]	250	V_{CC}	I-PU	IOD-PU O	I-PU TS-PD		5	
MA0 {CFG0}	C10	O {I}	C–E[3]	70	V_{CC}	I-PD	O	TS-PPD		6	
MA1 {CFG1}	B11	O {I}	C–E[3]	70	V_{CC}	I-PD	O	TS-PPD		6	
MA2 {CFG2}	A12	O {I}	C–E[3]	70	V_{CC}	I-PD	O	TS-PPD		6, 7	
MA3 {CFG3}	B12	O {I}	C–E[3]	70	V_{CC}	I-PD	O	TS-PPD		6	
MA4	C11	O	C–E[3]	70	V_{CC}	I-PD	O	TS-PD		6, 8	
MA5	A14	O	C–E[3]	70	V_{CC}	L	O	TS-PD		8	
MA6	B13	O	C–E[3]	70	V_{CC}	L	O	TS-PD		8	
MA7	C12	O	C–E[3]	70	V_{CC}	L	O	TS-PD		8	
MA8	A15	O	C–E[3]	70	V_{CC}	L	O	TS-PD		8	
MA9	B14	O	C–E[3]	70	V_{CC}	L	O	TS-PD		8	

Table 7. Pin State Table—Memory Interface[1] (Continued)

Signal Name [Alternate Function]	Pin #	Type	Output Drive	Max Load (pF)	Supply	Reset State	Normal Operation	Suspend State	Power Down Group	Note	5 V
MA10	C13	O	C–E[3]	70	V_{CC}	L	O	TS-PD		8	
MA11	A16	O	C–E[3]	70	V_{CC}	L	O	TS-PD		8	
$\overline{MWE}$	A11	O	C–E[3]	70	V_{CC}	H	O	H[TS-PD]	A	2	
$\overline{RAS0}$	C15	O	C–E[3]	50	V_{CC}	H	O	O[L][TS-PD]	A	2	
$\overline{RAS1}$	D14	O	C–E[3]	50	V_{CC}	H	O	O[L][TS-PD]	A	2	
$\overline{ROMCS0}$	R18	O	B	50	V_{CC}	H	O	H[TS-PD][TS]	B	9	S
$\overline{ROMCS1}$	T19	O	B	50	V_{CC}	H	O	H[TS-PD][TS]	B	9	S
$\overline{ROMRD}$	V20	O	B	50	V_{CC}	H	O	H[TS-PD][TS]	B	9	S
$\overline{ROMWR}$	U19	O	B	50	V_{CC}	H	O	H[TS-PD][TS]	B	9	S

Notes:

1. Pin states for D31–D16 are listed in Table 6 on page 42.

2. $\overline{RAS3}$–$\overline{RAS0}$, $\overline{CASH3}$–$\overline{CASH0}$, $\overline{CASL3}$–$\overline{CASL0}$, and $\overline{MWE}$ Suspend state of the pins:

–The $\overline{RAS}$ and $\overline{CAS}$ signals remain active if the DRAM interface is configured for $\overline{CAS}$-before-$\overline{RAS}$ refresh in Suspend mode.

–The $\overline{RAS}$ and $\overline{CAS}$ signals will be Low if the DRAM is configured for self-refresh in Suspend mode.

–Will be three-stated with a pulldown resistor if the DRAM interface is programmed to be disabled so the DRAM can be powered down (Power-Down Group A).

–Will not be affected by this when the $\overline{RAS}$ and $\overline{CAS}$ signals that share pins with other functions ($\overline{RAS3}$–$\overline{RAS2}$, $\overline{CASH3}$–$\overline{CASH2}$, and $\overline{CASL3}$–$\overline{CASL2}$) are not enabled to come out of the chip.

–The $\overline{MWE}$ signal will be driven out High (deasserted) when the DRAM is programmed to be left powered (Power-Down Group A).

***Summary:** These pins have built-in pulldown resistors that are invoked by:*

–Suspend mode and DRAM interface programmed for power-down in Suspend (Power-Down Group A), and the pins are enabled as RAS/CAS for $\overline{RAS3}$–$\overline{RAS2}$, $\overline{CASH3}$–$\overline{CASH2}$, and $\overline{CASL3}$–$\overline{CASL2}$.

3. C, D, and E output drives are programmable.

4. The data bus D15–D0 has built-in pulldown resistors that are invoked when the data bus signals are inputs.

5. Memory Address MA12 Suspend state of the pin:

Will be three-stated with a pulldown resistor. This will work for $\overline{CAS}$-before-$\overline{RAS}$ refresh, self-refresh, and the DRAM powered down.

***Summary:** This pin has a built-in pulldown resistor that is invoked by Suspend mode.*

6. Memory Address MA4–MA0 pins are shared with the power-on configuration signals so the reset state of the pins has a pull-down resistor on these signals.

This default configuration will choose: not test mode and an 8-bit ROM/Flash memory accessed by $\overline{ROMCS0}$ with the SD buffer-control signals disabled. The pulldown resistors are from 50 K to 150 K; they need to be overridden by pullup resistors on the board if other configurations are needed.

These pulldown resistors are disabled after reset; they are not active during normal chip operation.

For configuration signals CFG0, CFG1, CFG2, and CFG3, if the system uses the default configuration, the pulldown resistors will be active again in Suspend mode. If external pullup resistors are used on the board for a different configuration, the pins with external pullups will three-state in Suspend mode without pulldown resistors.

The reserved signal on MA4 is only used for AMD testing; it should not be pulled up on the system design. This pin will always go to three-state with a pulldown resistor in Suspend mode.

***Summary:** Each pin has a built-in pulldown resistor that is invoked by:*

–Reset

–Suspend mode and the configuration pin being Low during reset (for CFG3–CFG0).

–Suspend mode for the reserved signal on MA4.

7. The CFG2 pin is not supported on the ÉlanSC410 microcontroller.

8. Memory Address MA11–MA4 Suspend state of the pins:

Will be three-stated with a pulldown resistor. This will work for $\overline{CAS}$-before-$\overline{RAS}$ refresh, self-refresh, and the DRAM powered down.

***Summary:** These pins have built-in pulldown resistors that are invoked by Suspend mode.*

9. The ROM control signals have three programmable options for Suspend mode:

–Driven High (inactive)

–Three-stated with no pullup or pulldown. This is useful when the ROM is at 5 V and left powered in Suspend. The board design should not drive 3.3-V signals into a 5-V device during Suspend, because this can waste power. The system designer could provide large pullup resistors to 5 V for each of these signals on the board if this configuration is programmed.

– Three-stated with pulldown resistors when suspended with the intent of powering off the ROMs (Power-Down Group B).

***Summary:** These pins have built-in pulldown resistors that are invoked by:*

–Suspend mode; and the ROM interface is programmed to be powered off in Suspend mode (Power-Down Group B).

Table 8. Pin State Table—GPIOs/Parallel Port/PC Card Socket B

Signal Name [Alternate Function]	Pin #	Type	Output Drive	Max Load (pF)	Supply	Reset State	Normal Operation	Suspend State	Power Down Group	Note	5 V
GPIO15 [PCMA_VPP2]	Y15	B [O]	B	50	V_{CC}	I-PD	I-PPD[O] O	I-PPD[O] O		1,2	
GPIO16 [PCMB_VCC]	V15	B [O]	B	50	V_{CC}	I-PD	I-PPD[O] O	I-PPD[O] O		1, 2	
GPIO17 [PCMB_VPP1]	W15	B [O]	B	50	V_{CC}	I-PD	I-PPD[O] O	I-PPD[O] O		1, 2	
GPIO18 [PCMB_VPP2]	Y14	B [O]	B	50	V_{CC}	I-PD	I-PPD[O] O	I-PPD[O] O		1, 2	
GPIO19 [LBL2]	V14	B [O]	B	50	V_{CC}	I-PU	I-PPU[O] O	I-PPU[O] O		1	
GPIO20 [CD_A2]	G18	B [I]	B	50	V_{CC}	I-PU	I-PPU[O] I-PPU	I-PPU[O] I-PPUD	G	1, 2	S
GPIO21 [PPDWE] *(PC Card Enabled)*	V3	B [O]	C	30	V_{CC}	I-PU	I-PPU[O] O TS-PD	I-PPU[O] H[TS-PD][TS] TS-PD	H	3	S
GPIO22 [PPOEN] *(PC Card Enabled)*	T3	B [O]	C	30	V_{CC}	I-PU	I-PPU[O] O TS-PD	I-PPU[O] H[TS-PD][TS] TS-PD	H	3	S
GPIO23 [SLCT] [WP_B]	U4	B [I] [I]	D	150	V_{CC}	I-PU	I-PPU[O] I-PU I-PPU	I-PPU[O] I-PU[I-PD] I-PPUD	H	2, 3	S
GPIO24 [BUSY] [BVD2_B]	U3	B [I] [I]	D	150	V_{CC}	I-PU	I-PPU[O] I-PU I-PPU	I-PPU[O] I-PU[I-PD] I-PPUD	H	2, 3	S
GPIO25 [ACK] [BVD1_B]	U2	B [I] [I]	D	150	V_{CC}	I-PU	I-PPU[O] I-PU I-PPU	I-PPU[O] I-PU[I-PD] I-PPUD	H	2, 3	S
GPIO26 [PE] [RDY_B]	W2	B [I] [I]	D	150	V_{CC}	I-PU	I-PPU[O] I-PU I-PPU	I-PPU[O] I-PU[I-PD] I-PPUD	H	2, 3	S
GPIO27 [ERROR] [CD_B]	V4	B [I] [I]	D	150	V_{CC}	I-PU	I-PPU[O] I-PU I-PPU	I-PPU[O] I-PU[I-PD] I-PPUD	H	2, 3	S
GPIO28 [INIT] [REG_B]	Y2	B [OD][O] [O]	D	150	V_{CC}	OD-PU	I-PPU[O] OD-PU[O] O	I-PPU[O] OD-PU[OD-PD] H[TS-PD][TS]	H	2, 3	S
GPIO29 [SLCTIN] [RST_B]	W3	B [OD][O] [O]	D	150	V_{CC}	OD-PU	I-PPU[O] OD-PU[O] O	I-PPU[O] OD-PU[OD-PD] L[TS-PD]	H	2, 3	S
GPIO30 [AFDT] [MCEH_B]	W1	B [OD][O] [O]	D	150	V_{CC}	OD-PU	I-PPU[O] OD-PU[O] O	I-PPU[O] OD-PU[OD-PD] H[TS-PD][TS]	H	2, 3	S
GPIO31 [STRB] [MCEL_B]	V2	B [OD][O] [O]	D	150	V_{CC}	OD-PU	I-PPU[O] OD-PU[O] O	I-PPU[O] OD-PU[OD-PD] H[TS-PD][TS]	H	2, 3	S
GPIO_CS13 [PCMA_VCC]	V16	B [O]	B	50	V_{CC}	I-PD	I-PPD[O] O	I-PPD[O] O		1, 2	
GPIO_CS14 [PCMA_VPP1]	W16	B [O]	B	50	V_{CC}	I-PD	I-PPD[O] O	I-PPD[O] O		1, 2	

Notes:

1. *The shared GPIO20–GPIO15, GPIO_CS14–GPIO_CS13, and PC Card battery signals:*

   As GPIO_CSxs, the signals are active in Suspend mode: that is, if they are inputs before Suspend, they are still inputs during Suspend (the GPIO_CSs can be used to wake up the system); if they are outputs before Suspend, they are still outputs during Suspend (the GPIO_CSs can be programmed to change state by mode). As inputs, the pullup or pulldown on the signal can be disabled; if disabled, it is disabled in all modes. When the signal is an output, the built-in resistors are automatically disabled.

   When enabled, the Latched Battery Low Detect function ($\overline{LBL2}$) that is shared on GPIO19 is an output in all modes; there are no pullup or pulldown resistors active.

   On the ÉlanSC400 microcontroller, the PC Card functions shared on these pins are programmable by PC Card socket; the pin multiplexing options are explained earlier in this document. For the PC Card power control ($\overline{PCMA\_VCC}$, PCMA_VPP1, PCMA_VPP2, $\overline{PCMB\_VCC}$, PCMB_VPP1, PCMB_VPP2), the signals are outputs for each mode.

   For the second Card Detect ($\overline{CD\_A2}$):

   –Reset invokes pullup.

   –During normal operation, the pullup resistor can be disabled by a register bit.

   –During Suspend mode, the input will have a pulldown if the PC Card Socket A interface is programmed to be powered off in Suspend mode (Power-Down Group G). If the socket is not programmed to be powered off in Suspend mode, the input will have the same state as when operating: the pullup is programmable to be enabled or not.

2. *The PC Card signals $\overline{MCEL\_B}$, $\overline{MCEH\_B}$, RST_B, $\overline{REG\_B}$, $\overline{CD\_B}$, $\overline{RDY\_B}$, $\overline{BVD1\_B}$, BVD2_B, WP_B, $\overline{CD\_A2}$, PCMB_VPP2, PCMB_VPP1, $\overline{PCMB\_VCC}$, PCMA_VPP1, PCMA_VPP2, and $\overline{PCMA\_VCC}$ are not supported on the ÉlanSC410 microcontroller.*

3. *The shared parallel port, PC Card Socket B control, and GPIO signals:*

   –These signals default to the GPIO interface on reset.

   –As a parallel port in Suspend mode, these signals are programmable to accommodate the parallel port powered up or down.

   –As PC Card control on the ÉlanSC400 microcontroller, these signals have the same features as the Socket A control signals.

   –As GPIOs, these signals are not handled specially in Suspend, they remain the same as they were when the chip was active (i.e., they remain as inputs with the pullup enabled or not, or continue to drive out the same value if they were outputs).

   ***Summary:** Shared parallel port/PC Card Socket B/GPIO signals: Built-in pullup and pulldown resistors that are invoked by:*

   –Reset invokes pullups

   –As parallel port signals:

   - *Operating: pullups are enabled if not EPP mode. Outputs without pullup or pulldowns if EPP mode.*
   - *Suspend: pullups are enabled, unless the parallel port is programmed to be powered off in Suspend mode, in which case pulldowns are enabled.*
   - *If EPP mode is enabled for the parallel port, the outputs are driven out at their last value in Suspend mode.*

   –As PC Card Socket B signals (ÉlanSC400 microcontroller only):

   - *Operating: outputs have no pullups or pulldowns; inputs have pullups that can be disabled by programming a bit.*
   - *Suspend: outputs are driven out inactive with no pullups or pulldowns unless the PC Card Socket B is programmed to be powered off in Suspend mode; then the outputs go to three-state with pulldown resistors; inputs will be the same as they were when operating, with a pullup resistor that can be disabled by programming, unless the PC Card Socket B is programmed to be powered off in Suspend mode (Power-Down Group H), in which case the inputs have pulldown resistors enabled.*

   –As GPIO signals:

   - *Operating or Suspend: as outputs they have no pullups or pulldowns; as inputs they have pullups that can be disabled by programming a bit; no change of state when the system goes to Suspend.*

Table 9. Pin State Table—GPIOs/ISA Bus

Signal Name [Alternate Function]	Pin #	Type	Output Drive	Max Load (pF)	Supply	Reset State	Normal Operation	Suspend State	Power Down Group	Note	5 V
GPIO_CS5 [IOCS16]	W19	B [I]	B	50	V_{CC}	I-PU	I-PPU[O] I-PU	I-PPU[O] I-PU[I-PD]	C	1	S
GPIO_CS6 [IOCHRDY]	V18	B [STI]	B	50	V_{CC}	I-PU	I-PPU[O] I-PU	I-PPU[O] I-PU[I-PD]	C	1	S
GPIO_CS7 [PIRQ1]	Y19	B [I]	B	50	V_{CC}	I-PU	I-PPU[O] I-PU	I-PPU[O] I-PU[I-PD]	C	1	S
GPIO_CS8 [PIRQ0]	W18	B [I]	B	50	V_{CC}	I-PU	I-PPU[O] I-PU	I-PPU[O] I-PU[I-PD]	C	1	S
GPIO_CS9 [TC]	Y18	B [O]	C	50	V_{CC}	I-PU	I-PPU[O] O	I-PPU[O] TS-PD		1	S
GPIO_CS10 [AEN]	V17	B [O]	C	50	V_{CC}	I-PU	I-PPU[O] O	I-PPU[O] TS-PD		1	S
GPIO_CS11 [PDACK0]	W17	B [O]	C	50	V_{CC}	I-PU	I-PPU[O] O	I-PPU[O] H[TS-PD][TS]	C	1	S
GPIO_CS12 [PDRQ0]	Y17	B [I]	B	50	V_{CC}	I-PD	I-PPD[O] I-PD	I-PPD[O] I-PD		1	S

Notes:

1. *The shared GPIO_CS12–GPIO_CS5 and ISA signals:*

   As GPIO_CS signals, they are active in Suspend mode: that is, if they are inputs before Suspend, they are still inputs during Suspend (they can be used to wake up the system); if they are outputs before Suspend, they are still outputs during Suspend (they can be programmed to change state by mode). As inputs, the pullup or pulldown on the signal can be disabled; if disabled, it is disabled in all modes. When the signal is an output, the built-in resistors are automatically disabled.

   The ISA function for each pin is programmable by functional group: that is, the system can choose to use PIRQ0 and still use the DMA pins as GPIO_CSxs (the pin multiplexing options are explained elsewhere in this document). As ISA signals, these pins are programmable to support a system with ISA peripherals powered up or down in Suspend mode (Power-Down Group C). For those signals that are High when deasserted, there is an option to three-state them with no built-in resistors, so an external resistor can be placed on the board to pull them up to 5 V.

   Summary: *GPIO_CS12: Built-in pulldown resistor that is invoked by:*

   –Reset

   –ISA signal enabled on this pin (the pin will be PDRQ0).

   –The pulldown is disabled by this pin being a GPIO_CS and an output.

   –The pulldown can be programmed to be disabled when the pin is a GPIO_CS input.

   Summary: *GPIO_CS11: Built-in pullup and pulldown resistors that are invoked by:*

   –Reset invokes the pullup.

   –When enabled as the ISA signal $\overline{PDACK0}$:

   - *In normal operation, this signal is an output and no pullup or pulldown is needed.*
   - *The pulldown is invoked by Suspend mode and the ISA bus is programmed to be powered off in Suspend (Power-Down Group C).*
   - *If the ISA bus is programmed for 5-V use and is not powered down in Suspend, then this signal is three-state without a built-in pullup or pulldown resistor.*

   –When enabled as the GPIO_CS11 signal:

   - *As an output, the pullup and pulldown are disabled in all modes, and these GPIO_CSx signals can be active in Suspend.*
   - *As an input, the pullup can be programmed to be enabled or disabled; this will then be the state of the pin in all modes, including Suspend.*

   Summary: *GPIO_CS10–GPIO_CS9: Built-in pullup and pulldown resistors that are invoked by:*

   –Reset invokes the pullups.

   –When enabled as the ISA signals AEN and TC:

   - *In normal operation, these signals are outputs and no pullup or pulldown is needed.*
   - *The pulldowns are invoked by Suspend mode.*

   –When enabled as the GPIO_CS10–GPIO_CS9 signals:

   - *As an output, the pullup and pulldown are disabled in all modes, and these GPIO_CS signals can be active in Suspend.*
   - *As an input, the pullup can be programmed to be enabled or disabled; this will then be the state of the pin in all modes, including Suspend.*

   Summary: *GPIO_CS8–GPIO_CS5: Built-in pullup and pulldown resistors that are invoked by:*

   –Reset invokes the pullups.

   –When enabled as PIRQ1–PIRQ0, IOCHRDY, and $\overline{IOCS16}$:

   - *In normal operation and Suspend, these signals are inputs and the pullup resistors are active.*
   - *The pulldowns are invoked by Suspend mode and the ISA bus interface programmed for power off in Suspend (Power-Down Group C).*

   –When enabled as the GPIO_CS8–GPIO_CS5 signals:

   - *As an output, the pullup and pulldown are disabled in all modes, and these GPIO_CSx signals can be active in Suspend.*
   - *As an input, the pullup can be programmed to be enabled or disabled; this will then be the state of the pin in all modes, including Suspend.*

Table 10. Pin State Table—GPIOs/System Data (SD) Buffer Control

Signal Name [Alternate Function]	Pin #	Type	Output Drive	Max Load (pF)	Supply	Reset State	Normal Operation	Suspend State	Power Down Group	Note	5 V
GPIO_CS2 [[DBUFRDL]]	C18	B [[O]]	C	50	V_{CC}	I-PU	I-PPU[O] O	I-PPU[O] TS-PD		1	
GPIO_CS3 [[DBUFRDH]]	D17	B [[O]]	C	50	V_{CC}	I-PU	I-PPU[O] O	I-PPU[O] TS-PD		1	
GPIO_CS4 [[DBUFOE]]	C4	B [[O]]	C	50	V_{CC}	I-PU	I-PPU[O] O	I-PPU[O] H[TS-PD][TS]	I	1	S
KBD_ROW13 [[R32BFOE]]	A3	STI-OD [O]	C	250	V_{CC}	I-PU	IOD-PU O	I-PU H[TS-PD][TS]	I	2	S

Notes:

1. *The data buffer control signals are shared with the GPIO_CS4–GPIO_CS2 signals and with the keyboard row signal:*

   When the data buffer control signals are enabled on the pins, they will drive inactive during Suspend mode, go three-state without resistors to allow an external resistor to 5 V, or three-state with a pulldown to support powering off the data buffer.

   Summary: *GPIO_CS4–GPIO_CS2/$\overline{\text{DBUFOE}}$/$\overline{\text{DBUFRDH}}$/$\overline{\text{DBUFRDL}}$: Built-in pullup and pulldown resistors that are invoked by:*

   –Reset invokes pullup.

   –When buffer control is invoked by the configuration pin, these pins are outputs without any pullups or pulldowns.

   –When buffer control is enabled and in Suspend mode, $\overline{\text{DBUFRDH}}$ and $\overline{\text{DBUFRDL}}$ are three-state with the pulldowns enabled; $\overline{\text{DBUFOE}}$ has three options:

   - *High (inactive) with no pullup or pulldown.*
   - *Three-state with a pulldown if it is programmed for the buffer to be powered off in Suspend mode (Power-Down Group I).*
   - *Three-state with no pulldown if it is programmed for the buffer to be powered on in Suspend mode and at 5 V.*

   –When enabled as the GPIO_CS4–GPIO_CS2 signals:

   - *As an output, the pullup and pulldown are disabled in all modes, and these GPIO_CS signals can be active in Suspend.*
   - *As an input, the pullup can be programmed to be enabled or disabled. This will then be the state of the pin in all modes, including Suspend.*

2. *This data buffer control signal ($\overline{\text{R32BFOE}}$) is shared with the keyboard row signal:*

   When the data buffer control signals are enabled on the pins, they will drive inactive during Suspend mode, go three-state without resistors to allow an external resistor to 5 V, or three-state with a pulldown to support powering off the data buffer.

   Summary: *KBD_ROW13/$\overline{\text{R32BFOE}}$: Built-in pullup and pulldown resistors that are invoked by:*

   –Reset invokes the pullup.

   –As $\overline{\text{R32BFOE}}$, this pin is an output without a pullup or pulldown.

   –When buffer control is enabled and in Suspend mode, $\overline{\text{R32BFOE}}$ has three options:

   - *High (inactive) with no pullup or pulldown.*
   - *Three-state with a pulldown if it is programmed for the buffer to be powered off in Suspend mode.*
   - *Three-state with no pulldown if it is programmed for the buffer to be powered on in Suspend mode and at 5 V.*

   –When enabled as the keyboard row signal, this signal has a pullup enabled at all times.

Table 11. Pin State Table—GPIOs

Signal Name [Alternate Function]	Pin #	Type	Output Drive	Max Load (pF)	Supply	Reset State	Normal Operation	Suspend State	Power Down Group	Note	5 V
GPIO_CS0	V19	B	B	50	V_{CC}	I-PU	I-PPU[O]	I-PPU[O]	E	1	S
GPIO_CS1	Y20	B	B	50	V_{CC}	I-PU	I-PPU[O]	I-PPU[O]	E	1	S

Notes:

1. *The GPIO_CS signals become inputs in Suspend mode with either a pullup resistor for devices that are left powered, or a pulldown resistor for devices that are to be powered off.*

   Summary: *GPIO_CS1–GPIO_CS0: Built-in pullup and pulldown resistors that are invoked by:*

   Reset invokes pullup.

   When enabled as the GPIO_CS1–GPIO_CS0 signals:

   As an output, the pullup and pulldown are disabled in all modes, and these GPIO_CSxs can be active in Suspend.

   As an input, the pullup can be programmed to be enabled or disabled; this will then be the state of the pin in all modes, including Suspend.

Table 12. Pin State Table—Serial Port

Signal Name [Alternate Function]	Pin #	Type	Output Drive	Max Load (pF)	Supply	Reset State	Normal Operation	Suspend State	Power Down Group	Note	5 V
$\overline{CTS}$	V10	I			V_{CC}	I-PU	I-PU [I-PD]	I-PU[I-PD]	D	1	
$\overline{DCD}$	V9	I			V_{CC}	I-PU	I-PU [I-PD]	I-PU[I-PD]	D	1	
$\overline{DSR}$	W10	I			V_{CC}	I-PU	I-PU [I-PD]	I-PU[I-PD]	D	1	
$\overline{DTR}$	Y8	O	A	30	V_{CC}	H	O [TS-PD]	TS-PD		1	
$\overline{RIN}$	V11	I			V_{CC}	I-PU	I-PU [I-PD]	I-PU[I-PD]	D	1	
$\overline{RTS}$	W8	O	A	30	V_{CC}	H	O [TS-PD]	TS-PD		1	
SIN	W11	I			V_{CC}	I-PU	I-PU [I-PD]	I-PU[I-PD]	D	1	
SOUT	Y10	O	A	30	V_{CC}	H	O [TS-PD]	TS-PD		1	

Notes:

1. *The serial port output signals are three-state with built-in pulldown resistors in Suspend mode. The serial port input signals can be left as inputs with pullups for a Suspend when the serial device is left powered. Or, they can be configured as inputs with pulldown resistors if the serial device is to be powered off (Power-Down Group D).*

   Summary: *The serial port output pins have built-in pulldown resistors that are invoked by Suspend mode.*

   Summary: *The serial port input pins have built-in pullup and pulldown resistors that are invoked by:*

   –Reset invokes the pullup resistors.

   –Operating: the pullup resistors are enabled.

   –Suspend mode invokes the pulldown resistors if the serial interface is programmed to be powered off in Suspend (Power-Down Group C); otherwise there are pullup resistors in Suspend mode.

Table 13. Pin State Table—Infrared Interface

Signal Name [Alternate Function]	Pin #	Type	Output Drive	Max Load (pF)	Supply	Reset State	Normal Operation	Suspend State	Power Down Group	Note	5 V
SIRIN	Y9	I			V_{CC}	I-PD	I-PPD	I-PPD		1	
SIROUT	V8	O	A	30	V_{CC}	L	O	TS-PD		1	

Notes:

1. *The serial infrared interface output and input settle to Suspend states that allow the device to be powered up or off. The output is three-state with a built-in pulldown resistor, and the input has a built-in pulldown resistor. The pulldown resistor on the input pin (SIRIN) can be programmed to be disabled during normal operation and Suspend mode.*

   Summary: *The serial infrared input pin has a built-in pulldown resistor that is invoked by:*

   –Reset invokes the pulldown resistor.

   –The pulldown resistor is then programmable to be there or not.

Table 14. Pin State Table—Keyboard Interface

Signal Name [Alternate Function]	Pin #	Type	Output Drive	Max Load (pF)	Supply	Reset State	Normal Operation	Suspend State	Power Down Group	Note	5 V
KBD_COL0 [XT_DATA]	E2	OD-STI [B]	D	250	V_{CC}	OD-PU	IOD-PPUD IOD-PPUD	IOD-PPUD IOD-PPUD		1	S
KBD_COL1 [XT_CLK]	D1	OD-STI [B]	D	250	V_{CC}	OD-PU	IOD-PPUD IOD-PPUD	IOD-PPUD IOD-PPUD		1	S
KBD_COL2/ PIRQ3	A2	OD-STI [I]	D	250	V_{CC}	OD-PU	IOD-PPUD I-PPUD	IOD-PPUD I-PPUD[I-PD]	C	1	S
KBD_COL3/ PIRQ4	B3	OD-STI [I]	D	250	V_{CC}	OD-PU	IOD-PPUD I-PPUD	IOD-PPUD I-PPUD[I-PD]	C	1	S
KBD_COL4/ PIRQ5	C3	OD-STI [I]	D	250	V_{CC}	OD-PU	IOD-PPUD I-PPUD	IOD-PPUD I-PPUD[I-PD]	C	1	S
KBD_COL5/ PIRQ6	A1	OD-STI [I]	D	250	V_{CC}	OD-PU	IOD-PPUD I-PPUD	IOD-PPUD I-PPUD[I-PD]	C	1	S
KBD_COL6/ PIRQ7	B2	OD-STI [I]	D	250	V_{CC}	OD-PU	IOD-PPUD I-PPUD	IOD-PPUD I-PPUD[I-PD]	C	1	S
KBD_COL7	C5	OD-STI	D	250	V_{CC}	OD-PU	IOD-PPUD	IOD-PPUD		1	
KBD_ROW0 [CASL2]	B19	STI-OD [O]	D	250	V_{CC}	I-PU	IOD-PU O	I-PU O[L][TS-PD]	A	2	
KBD_ROW1 [CASL3]	D16	STI-OD [O]	D	250	V_{CC}	I-PU	IOD-PU O	I-PU O[L][TS-PD]	A	2	
KBD_ROW2 [CASH2]	C17	STI-OD [O]	D	250	V_{CC}	I-PU	IOD-PU O	I-PU O[L][TS-PD]	A	2	
KBD_ROW3 [CASH3]	B18	STI-OD [O]	D	250	V_{CC}	I-PU	IOD-PU O	I-PU O[L][TS-PD]	A	2	
KBD_ROW4 [RAS2]	D15	STI-OD [O]	P-C,E	250	V_{CC}	I-PU	IOD-PU O	I-PU O[L][TS-PD]	A	2	
KBD_ROW5 [RAS3]	C16	STI-OD [O]	P-C,E	250	V_{CC}	I-PU	IOD-PU O	I-PU O[L][TS-PD]	A	2	
KBD_ROW6 [MA12]	B17	STI-OD [O]	P-C,E	250	V_{CC}	I-PU	IOD-PU O	I-PU TS-PD		3	
KBD_ROW7 [PDACK1]	D3	STI-OD [O]	C	250	V_{CC}	I-PU	IOD-PU O	I-PU H[TS-PD][TS]	C		S
KBD_ROW8 [PDRQ1]	C2	STI-OD [I]	C	250	V_{CC}	I-PU	IOD-PU I-PD	I-PU I-PD			S
KBD_ROW9 [PIRQ2]	E3	STI-OD [I]	C	250	V_{CC}	I-PU	IOD-PU I-PU	I-PU I-PU[I-PD]	C		S
KBD_ROW10 [BALE]	D2	STI-OD [O]	C	250	V_{CC}	I-PU	IOD-PU O	I-PU TS-PD			S
KBD_ROW11 [SBHE]	C1	STI-OD [O]	C	250	V_{CC}	I-PU	IOD-PU O	I-PU H[TS-PD][TS]	C		S
KBD_ROW12 [MCS16]	F3	STI-OD [I]	C	250	V_{CC}	I-PU	IOD-PU I-PU	I-PU I-PU[I-PD]	C		S
KBD_ROW13 [[R32BFOE]]	A3	STI-OD [O]	C	250	V_{CC}	I-PU	IOD-PU O	I-PU H[TS-PD][TS]	I	4	S

Notes:

1. *The keyboard column signals are shared with the programmable IRQs and XT keyboard signals.*

   As keyboard column signals and XT keyboard signals, they are inputs and open drain outputs with pullup or pulldown resistors in normal operation and Suspend mode. Each column signal is individually programmable for the pullup or pulldown in the keyboard extended registers.

   As IRQs, the pins are inputs with built-in pullup or pulldown resistors (use the same registers in the keyboard controller to enable pullups or pulldowns). During Suspend mode, they stay as inputs with the pullup or pulldown. Or, if Power-Down Group C is enabled for the ISA bus to be powered down in Suspend mode and a bit is set identifying that these signals are being used as IRQs, they will have pulldown resistors activated. There is no programmable bit to make these signals IRQs beyond the extended register in the interrupt controller that maps the pin to a particular IRQ. If the system must use any of these as IRQs, a bit must be set, notifying the chip, so that they can have the pulldown resistors invoked in Suspend mode.

   Summary: *As keyboard column and XT keyboard signals:*

   – *Pullup or pulldown resistor depending on the setting of the Keyboard Column Pullup/Pulldown register in the keyboard controller.*

   Summary: *As programmable IRQ signals:*

   – *Pullup or pulldown resistors during normal operation and Suspend (depending on the configuration register in the keyboard controller.*

   – *Pulldown resistors during suspend if Power-Down Group C (the ISA bus) is enabled for power-down in Suspend, and a bit is set indicating that these signals are used as IRQs and need to be pulled down in Suspend.*

2. $\overline{RAS3}$–$\overline{RAS2}$, $\overline{CASH3}$–$\overline{CASH2}$, and $\overline{CASL3}$–$\overline{CASL2}$ *Suspend state of the pins:*

   – *The* $\overline{RAS}$ *and* $\overline{CAS}$ *signals remain active if the DRAM interface is configured for* $\overline{CAS}$*-before-*$\overline{RAS}$ *refresh in Suspend mode.*

   – *The* $\overline{RAS}$ *and* $\overline{CAS}$ *signals will be Low if the DRAM is configured for self-refresh in Suspend mode.*

   – *Will be three-stated with a pulldown resistor if the DRAM interface is programmed to be disabled so the DRAM can be powered down (Power-Down Group A).*

   – *Will not be affected by this when the* $\overline{RAS}$ *and* $\overline{CAS}$ *signals that share pins with other functions are not enabled to come out of the chip.*

   Summary: *These pins have built-in pulldown resistors that are invoked by:*

   – *Suspend mode and DRAM interface programmed for power-down in Suspend (Power-Down Group A), and the pins are enabled as* $\overline{RAS}$/$\overline{CAS}$ *for* $\overline{RAS3}$–$\overline{RAS2}$, $\overline{CASH3}$–$\overline{CASH2}$, *and* $\overline{CASL3}$–$\overline{CASL2}$.

3. *Memory Address MA12 Suspend state of the pin:*

   Will be three-stated with a pulldown resistor. This will work for $\overline{CAS}$*-before-*$\overline{RAS}$ *refresh, self-refresh, and the DRAM powered down.*

   Summary: *This pin has a built-in pulldown resistor that is invoked by Suspend mode.*

4. *The data buffer control signal* $\overline{R32BFOE}$ *that is shared with the keyboard row signal:*

   When the data buffer control signals are enabled on the pins, they will drive inactive during Suspend mode, go three-state without resistors to allow an external resistor to 5 V, or three-state with a pulldown to support powering off the data buffer.

   Summary: *KBD_ROW13/*$\overline{R32BFOE}$*: Built-in pullup and pulldown resistors that are invoked by:*

   – *Reset invokes the pullup.*

   – *As* $\overline{R32BFOE}$*, this pin is an output without a pullup or pulldown.*

   – *When buffer control is enabled and in Suspend mode,* $\overline{R32BFOE}$ *has three options:*

   - *High (inactive) with no pullup or pulldown.*
   - *Three-state with a pulldown if it is programmed for the buffer to be powered off in Suspend mode.*
   - *Three-state with no pulldown if it is programmed for the buffer to be powered on in Suspend mode and at 5 V.*

   – *When enabled as the keyboard row signal, this signal has a pullup enabled at all times.*

Table 15. Pin State Table—PC Card Socket A

Signal Name [Alternate Function]	Pin #	Type	Output Drive	Max Load (pF)	Supply	Reset State	Normal Operation	Suspend State	Power Down Group	Note	5 V
BVD1_A	V1	I			V_{CC}	I-PU	I-PPU	I-PPUD	G	1, 2	S
BVD2_A	R3	I			V_{CC}	I-PU	I-PPU	I-PPUD	G	1, 2	S
$\overline{\text{CD\_A}}$	R2	I			V_{CC}	I-PU	I-PPU	I-PPUD	G	1, 2	S
ICDIR	M3	O	B	50	V_{CC}	L	O	H[TS-PD][TS]	G	1, 2	S
$\overline{\text{MCEH\_A}}$ [[BNDSCN_TMS]]	N3	O [[I]]	B	50	V_{CC}	H	O [I-PD]	H[TS-PD][TS]	G	1, 2	S
$\overline{\text{MCEL\_A}}$ [[BDNSCN_TCK]]	P2	O [[I]]	B	50	V_{CC}	H	O [I-PD]	H[TS-PD][TS]	G	1, 2	S
$\overline{\text{OE}}$	P1	O	B	50	V_{CC}	H	O	H[TS-PD][TS]	G	1, 2	S
$\overline{\text{RDY\_A}}$	P3	I			V_{CC}	I-PU	I-PPU	I-PPUD	G	1, 2	S
$\overline{\text{REG\_A}}$ [[BNDSCN_TDO]]	M2	O [[O]]	B	50	V_{CC}	H	O [O]	H[TS-PD][TS]	G	1, 2	S
RST_A [[BNDSCN_TDI]]	R1	O [[I]]	B	50	V_{CC}	O	O [I-PD]	L[TS-PD]	G	1, 2	S
$\overline{\text{WAIT\_AB}}$	U1	I			V_{CC}	I-PU	I-PPU	I-PPUD	G	1, 2	S
$\overline{\text{WE}}$	N2	O	B	50	V_{CC}	H	O	H[TS-PD][TS]	G	1, 2	S
WP_A	T2	I			V_{CC}	I-PU	I-PPU	I-PPUD	G	1, 2	S

Notes:

1. *On the ÉlanSC400 microcontroller only, the PC Card control signals for Socket A:*

   The pullup resistors for the input signals are built in and can be disabled if external pullups are necessary (the external pullups can be on a different power plane).

   In Suspend mode, the signals can be configured for: a card not plugged in (inputs terminated with internal resistors), a card plugged in and powered (the output signals drive out inactive), a card plugged in and powered and at 5 V (the inactive High output signals are three-stated and pullup resistors should be put on the board), and a card plugged in and powered off (the signals terminated with pulldown resistors) (Power-Down Group G).

   Summary: *The outputs are built-in pulldown resistors that are invoked by:*

   –Suspend and PC Card Socket A is programmed to be powered off in Suspend (Power-Down Group G).

   –These are not pulldowns for normal operation. These are driven outputs.

   Summary: *The inputs are built-in pullup and pulldown resistors that are invoked by:*

   –Reset invokes pullups.

   –During normal operation, the pullup resistors can be disabled by a register bit.

   –During Suspend mode, the inputs will have pulldowns if the PC Card Socket A interface is programmed to be powered off in Suspend mode (Power-Down Group G). If the socket is not programmed to be powered off in Suspend mode, the inputs have the same state as when operating: the pullups are programmable to be enabled or not.

2. *The PC Card signals $\overline{\text{MCEL\_A}}$, $\overline{\text{MCEH\_A}}$, RST_A, $\overline{\text{REG\_A}}$, $\overline{\text{CD\_A}}$, $\overline{\text{RDY\_A}}$, BVD1_A, BVD2_A, WP_A, $\overline{\text{WAIT\_AB}}$, $\overline{\text{OE}}$, $\overline{\text{WE}}$, and ICDIR are not supported on the ÉlanSC410 microcontroller.*

Table 16. Pin State Table—Graphics Controller/VESA Local Bus Control

Signal Name [Alternate Function]	Pin #	Type	Output Drive	Max Load (pF)	Supply	Reset State	Normal Operation	Suspend State	Power Down Group	Note	5 V
FRM [VL_LCLK]	E19	O [O]	E	150	V_{CC}	TS-PD	O[TS-PD] O[TS-PD]	TS-PD TS-PD	F	1, 2	S
LC [VL_BE1]	E20	O [O]	D	150	V_{CC}	TS-PD	O[TS-PD] O[TS-PD]	TS-PD H[TS-PD]	F	1,2	S
LCDD0 [VL_RST]	B20	O [O]	D	150	V_{CC}	TS-PD	O[TS-PD] O[TS-PD]	TS-PD H[TS-PD]	F	1,2	S
LCDD1 [VL_ADS]	C19	O [O]	D	150	V_{CC}	TS-PD	O[TS-PD] O[TS-PD]	TS-PD H[TS-PD]	F	1,2	S
LCDD2 [VL_W/$\overline{R}$]	D18	O [O]	D	150	V_{CC}	TS-PD	O[TS-PD] O[TS-PD]	TS-PD H[TS-PD]	F	1,2	S
LCDD3 [VL_M/$\overline{IO}$]	C20	O [O]	D	150	V_{CC}	TS-PD	O[TS-PD] O[TS-PD]	TS-PD H[TS-PD]	F	1,2	S
LCDD4 [VL_LRDY]	D19	O [I]	D	150	V_{CC}	TS-PD	TS-PD[O] I	TS-PD I[I-PD]	F	1,2	S
LCDD5 [VL_D/$\overline{C}$]	E18	O [O]	D	150	V_{CC}	TS-PD	O[TS-PD] O[TS-PD]	TS-PD H[TS-PD]	F	1,2	S
LCDD6 [VL_LDEV]	F17	O [I]	D	150	V_{CC}	TS-PD	TS-PD[O] I	TS-PD I[I-PD]	F	1,2	S
LCDD7 [VL_BE3]	D20	O [O]	D	150	V_{CC}	TS-PD	O[TS-PD] O[TS-PD]	TS-PD H[TS-PD]	F	1,2	S
$\overline{LVDD}$ [VL_BLAST]	A19	O [O]	D	50	V_{CC}	TS-PU	O[TS-PU] O[TS-PU]	H H[TS-PD]	F	1,2	
$\overline{LVEE}$ [VL_BRDY]	A20	O [I]	D	50	V_{CC}	TS-PU	O[TS-PU] I	H I[I-PD]	F	1,2	
M [VL_BE2]	F18	O [O]	D	150	V_{CC}	TS-PD	O[TS-PD] O[TS-PD]	TS-PD H[TS-PD]	F	1,2	S
SCK [VL_BE0]	F19	O [O]	D	150	V_{CC}	TS-PD	O[TS-PD] O[TS-PD]	TS-PD H[TS-PD]	F	1,2	S

Notes:

1. *The shared graphics controller interface and VESA local bus pins:*

   These signals default to three-state with pulldown resistors and remain this way until an LCD or VL-bus interface is selected (all except $\overline{LVEE}$ and $\overline{LVDD}$).

   When the graphics controller is enabled on the ÉlanSC400 microcontroller, the signals will be three-state with pulldowns whenever the LCD is not enabled. This allows the LCD to be powered off in any mode, and prevents damage to the LCD by having it powered when the timing of the signals is not correct. In Suspend these signals are three-state with pulldowns. The LCD cannot be driven in Suspend.

   When the VESA local bus interface is enabled, the signals will become the inputs and outputs necessary for VL-bus support. In Suspend, the signals support leaving the VL device powered on or off (Power-Down Group F).

   ***Summary:** LCD control signals/VESA local bus control signals (all except $\overline{LVEE}$ and $\overline{LVDD}$) have built-in pulldown resistors that are invoked by:*

   –Reset invokes pulldowns.

   –Graphics controller disabled and VL-bus disabled invokes pulldowns.

   –VL-bus enabled and VL interface programmed for power-down in Suspend mode invokes pulldowns (Power-Down Group F).

   –Graphics controller enabled and LCD enabled. All pins are outputs with no termination.

   –Graphics controller enabled and LCD disabled. All pins are three-state with pulldowns.

   –VL-bus enabled and not Suspend mode. No pulldowns enabled.

   ***Summary:** $\overline{LVEE}$ and $\overline{LVDD}$ have built-in pullup and pulldown resistors that are invoked by:*

   –Reset invokes pullups.

   –VL-bus enabled and VL interface programmed for power-down in Suspend mode invokes pulldowns in Suspend mode (Power-Down Group F).

   –Graphics enabled. Drive out High in Suspend.

   –Graphics enabled. Both pins are outputs without pullups or pulldowns.

   –VL-bus enabled. No pullups or pulldowns in normal operation.

2. *The graphics controller signals LCDD7–LCDD0, M, LC, SCK, FRM, $\overline{LVEE}$, and $\overline{LVDD}$ are not supported on the ÉlanSC410 microcontroller.*

Table 17. Pin State Table—Miscellaneous

Signal Name [Alternate Function]	Pin #	Type	Output Drive	Max Load (pF)	Supply	Reset State	Normal Operation	Suspend State	Power Down Group	Note	5 V
32KXTAL1	Y6				VRTC					1	
32KXTAL2	Y4				VRTC					1	
ACIN	W12	STI			V_{CC}	I-PD	I-PD	I-PD			
BBATSEN	V6	A			VRTC	I	I	I		1	
$\overline{BL1}$	Y13	STI			V_{CC}	I	I	I			
$\overline{BL2}$	W13	STI			V_{CC}	I	I	I			
$\overline{BLO}$ [CLK_IO]	W14	STI[B]	B	50	V_{CC}	I	I I[O]	I I[TS-PD]			
BNDSCN_EN	Y11	I			V_{CC}	I-PD	I-PD	I-PD			
LF_HS	V5	A			AV_{CC}	Analog	Analog	Analog			
LF_INT	Y3	A			AV_{CC}	Analog	Analog	Analog			
LF_LS	W4	A			AV_{CC}	Analog	Analog	Analog			
LF_VID	W5	A			AV_{CC}	Analog	Analog	Analog		2	
$\overline{RESET}$	Y7	STI			V_{CC}	I	I	I		1	
SPKR	V7	O	B	50	V_{CC}	L	O	TS-PD			
SUS_RES/ KBD_ROW14	Y12	STI/STI			V_{CC}	I	I	I			

Notes:

1. *The 32-kHz crystal signals are active in all modes.*

   The $\overline{RESET}$ signal is enabled as an input in all modes to reset the whole chip.

   The BBATSEN signal is active during reset to sense the state of the backup battery.

   Summary: *No pullups or pulldowns on these pins.*

2. *The LF_VID signal is not supported on the ÉlanSC410 microcontroller.*

Table 18. Pin State Table—Power[1] and Ground

Signal Name (Alternate Function)	Pin #	Type
GND	D4	—
GND	D5	—
GND	D6	—
GND	D7	—
GND	D8	—
GND	D9	—
GND	D10	—
GND	D12	—
GND	D11	—
GND	E4	—
GND	F4	—
GND	G4	—
GND	H4	—
GND	H8	Thermal
GND	H9	Thermal
GND	H10	Thermal
GND	H11	Thermal
GND	H12	Thermal
GND	H13	Thermal
GND	J4	—
GND	J8	Thermal
GND	J9	Thermal
GND	J10	Thermal
GND	J11	Thermal
GND	J12	Thermal
GND	J13	Thermal
GND	K8	Thermal
GND	K9	Thermal
GND	K10	Thermal
GND	K11	Thermal
GND	K12	Thermal
GND	K13	Thermal
GND	L8	Thermal
GND	L9	Thermal
GND	L10	Thermal
GND	L11	Thermal
GND	L12	Thermal
GND	L13	Thermal
GND	M8	Thermal
GND	M9	Thermal
GND	M10	Thermal
GND	M11	Thermal
GND	M12	Thermal

Table 18. Pin State Table—Power[1] and Ground (Continued)

Signal Name (Alternate Function)	Pin #	Type
GND	M13	Thermal
GND	N8	Thermal
GND	N9	Thermal
GND	N10	Thermal
GND	N11	Thermal
GND	N12	Thermal
GND	N13	Thermal
GND	R17	—
GND	T17	—
GND	U5	—
GND	U6	—
GND	U7	—
GND	U8	—
GND	U9	—
GND	U10	—
GND	U11	—
GND	U12	—
GND	U13	—
GND	U14	—
GND	U15	—
GND	U16	—
GND	U17	—
GND_A	Y5	Analog
V_{CC}	A8	I/O
V_{CC}	A13	I/O
V_{CC}	A18	I/O
V_{CC}	B1	I/O
V_{CC}	B4	I/O
V_{CC}	B16	I/O
V_{CC}	E1	I/O
V_{CC}	E17	I/O
V_{CC}	G17	I/O
V_{CC}	H1	I/O
V_{CC}	J17	I/O
V_{CC}	J3	I/O
V_{CC}	L17	Logic
V_{CC}	M1	I/O
V_{CC}	M17	Logic
V_{CC}	M19	I/O
V_{CC}	N17	Logic
V_{CC}	P17	Logic
V_{CC}	P18	I/O
V_{CC}	T1	I/O
V_{CC}	U18	I/O

Table 18. Pin State Table—Power[1] and Ground (Continued)

Signal Name (Alternate Function)	Pin #	Type
V_{CC}	V13	I/O
V_{CC}_RTC	W7	RTC
V_{CC}	W9	I/O
V_{CC}	Y1	I/O
V_{CC}	Y16	I/O
V_{CC}_A	W6	Analog
V_{CC}_CPU	K4	CPU
V_{CC}_CPU	L4	CPU
V_{CC}_CPU	M4	CPU
V_{CC}_CPU	N4	CPU
V_{CC}_CPU	P4	CPU
V_{CC}_CPU	R4	CPU
V_{CC}_CPU	T4	CPU

Notes:

1. *See the signal descriptions under the Reset and Power subheading in the Signal Description table beginning on page 62 for additional information about the V_{CC} pins.*

SIGNAL DESCRIPTIONS

The descriptions in Table 19 are organized in alphabetical order within the functional group listed here.

Table 19. Signal Description Table

Signal	Type	Description
System Interface		
AEN	O	**DMA Address Enable** indicates that the current address active on the SA25–SA0 address bus is a memory address, and that the current cycle is a DMA cycle. All I/O devices should use this signal in decoding their I/O addresses, and should not respond when this signal is asserted. When AEN is asserted, the $\overline{\text{PDACK1}}$– $\overline{\text{PDACK0}}$ signals are used to select the appropriate I/O device for the DMA transfer. AEN is also asserted when a DMA cycle is occurring internal to the chip. On the ÉlanSC400 microcontroller, AEN is also asserted for all accesses to the PC Card I/O space to prevent ISA devices from responding to the $\overline{\text{IOR}}$/$\overline{\text{IOW}}$ signal assertions because these signals are shared between the PC Card and ISA interfaces.
BALE	O	**Bus Address Latch Enable** is driven at the beginning of an ISA bus cycle with a valid address. This signal can be used by external devices to latch the address for the current cycle. BALE is also asserted for all accesses to the PC Card interfaces (memory or I/O) (ÉlanSC400 microcontroller only) and all DMA cycles. This prevents an ISA device from responding to a cycle based on a previously latched address.
$\overline{\text{DBUFOE}}$	O	**Data Buffer Output Enable** controls the output enable on the external transceiver required to drive the peripheral data bus in local bus and 32-bit DRAM modes.
$\overline{\text{DBUFRDH}}$	O	**High Byte Data Buffer Direction Control** controls direction of data flow through the external transceiver required to drive the peripheral data bus in local bus and 32-bit DRAM mode. This is the control signal for the upper 8 bits of the data bus.
$\overline{\text{DBUFRDL}}$	O	**Low Byte Data Buffer Direction Control** controls direction of data flow through the external transceiver required to drive the peripheral data bus in local bus and 32-bit DRAM mode. This is the control signal for the lower 8 bits of the data bus.
IOCHRDY	STI PU	**I/O Channel Ready** should be driven by open-drain devices. When pulled Low during an ISA access, wait states are inserted in the current cycle. This pin has an internal weak pullup that should be supplemented by a stronger external pullup (usually 4.7 KΩ to 1 KΩ) for faster rise time.
$\overline{\text{IOCS16}}$	I	**I/O Chip Select 16:** The targeted I/O device drives this signal active early in the cycle to request a 16-bit transfer.
$\overline{\text{IOR}}$	O	**I/O Read Command** indicates that the current cycle is a read from the currently addressed I/O device. When this signal is asserted, the selected I/O device can drive data onto the data bus. This signal is also shared with the PC Card interface on the ÉlanSC400 microcontroller.
$\overline{\text{IOW}}$	O	**I/O Write Command** indicates that the current cycle is a write to the currently addressed I/O device. When this signal is asserted, the selected I/O device can latch data from the data bus. This signal is also shared with the PC Card interface on the ÉlanSC400 microcontroller.
$\overline{\text{MCS16}}$	I	**Memory Chip Select 16** indicates to the ISA control logic that the targeted memory device is a 16-bit-wide device.

Table 19. Signal Description Table (Continued)

Signal	Type	Description
MEMR	O	**Memory Read Command** indicates that the current cycle is a read of the currently addressed memory device. When this signal is asserted, the memory device can drive data onto the data bus.
MEMW	O	**Memory Write Command** indicates that the current cycle is a write of the currently addressed memory device. When this signal is asserted, the memory device can latch data from the data bus.
PDACK1–PDACK0	O	**Programmable DMA Acknowledge** signals can each be mapped to one of the seven available DMA channels. They are driven active (Low) back to the DMA initiator to acknowledge the corresponding DMA requests.
PDRQ1–PDRQ0	I	**Programmable DMA Requests** can each be mapped to one of the seven available DMA channels. They are asserted active (High) by a DMA initiator to request DMA service from the DMA controller.
PIRQ7–PIRQ0	I	**Programmable Interrupt Requests** can each be mapped to one of the available 8259 interrupt channels. They are asserted when a peripheral requires interrupt service. (Rising Edge/Active High Trigger)
RSTDRV	O	**System Reset** is the ISA bus reset signal. When this signal is asserted, all connected devices reinitialize to their reset state. This signal should not be confused with the internal CPU RESET and SRESET signals.
SA25–SA0	O	**System Address Bus** outputs the physical memory or I/O port latched addresses. It is used by all external peripheral devices other than main system DRAM. In addition, this is the local address bus in local bus mode.
SBHE	O	**System Byte High Enable** is driven active when the high data byte is to be transferred on the upper 8 bits of the ISA data bus.
SD15–SD0	B	**System Data Bus** is shared between ISA, 8- or 16-bit ROM/Flash memory, and PC Card peripherals (on the ÉlanSC400 microcontroller only) and can be directly connected to all of these devices. In addition, these signals are the upper word of the local data bus, the 32-bit DRAM interface, and the 32-bit ROM interface. In these modes, the system data bus can be generated via an external buffer connected to the SD bus and controlled by the buffer control signals provided.
SPKR	O	**Speaker, Digital Audio Output** controls an external speaker driver. It is generated from the internal 8254-compatible timer Channel 2 output ANDed with I/O Port 0061h[1] (Speaker Data Enable); on the ÉlanSC400 microcontroller, the PC Card speaker signals are exclusively ORed with each other and the speaker control function of the timer to generate the SPKR signal.
TC	O	**Terminal Count** is driven from the DMA controller pair to indicate that the transfer count for the currently active DMA channel has reached zero, and that the current DMA cycle is the last transfer.
Configuration Pins		
BNDSCN_EN	I	**Boundary Scan Enable** enables the boundary scan pin functions. When this pin is High, the boundary scan interface is enabled. When this pin is Low, the boundary scan pin functions are disabled and the pins are configured to their default functions. This pin must be held Low during reset for normal operation.
CFG1–CFG0	I	**Configuration Pins 1–0** select the data bus width for the physical device(s) selected by the ROMCS0 pin (i.e., 8-, 16-, or 32-bit-wide). These pins are sampled at the deassertion of RESET.
CFG2	I	**Configuration Pin 2** selects whether or not the system will boot from PC Card Socket A memory card or from the device attached to ROMCS0. This pin is sampled at the deassertion of RESET. This pin is not supported on the ÉlanSC410 microcontroller.
CFG3	I	**Configuration Pin 3** enables the SD buffer control signals, DBUFOE, DBUFRDH, and DBUFRDL. This pin is sampled at the deassertion of RESET.

Table 19. Signal Description Table (Continued)

Signal	Type	Description
Memory Interface		
CASH3–CASH0	O	**Column Address Strobe High** indicates to the DRAM devices that a valid column address is asserted on the MA lines. These CAS signals are for the odd banks (Banks 1 and 3); CASH3–CASH2 are for the high word; and CASH1–CASH0 is for the low word.
CASL3–CASL0	O	**Column Address Strobe Low** indicates to the DRAM devices that a valid column address is asserted on the MA lines. These CAS signals are for the even banks (Banks 0 and 2); CASL1–CASL0 are for the low word; CASL3–CASL2 are for the high word.
D31–D0	B	**Data Bus** is used for DRAM and local bus cycles. This bus is also used when interfacing to 32-bit ROMs.
MA12–MA0	O	**Memory Address:** The DRAM row and column addresses are multiplexed onto this bus. Row addresses are driven onto this bus and are valid upon the falling edge of RAS. Column addresses are driven onto this bus and are valid upon the falling edge of CAS.
MWE	O	**Write Enable** indicates an active write cycle to the DRAM devices. This signal is also used to three-state EDO DRAMs at the end of EDO read cycles.
R32BFOE	O	**ROM 32-Bit Buffer Output Enable** provides the buffer enable signal for the external transceivers on the low word of the ROM interface. This signal is automatically provided when the ROMCS0 interface is configured as 32 bit (the configuration can be done using either CFG1–CFG0 or CSC index 20h[1–0]). Once ROMCS0 is configured as 32 bit, all accesses to 32-bit ROM devices on ROMCS2–ROMCS0 result in the assertion of the R32BFOE signal.
RAS3–RAS0	O	**Row Address Strobe** indicates to the DRAM devices that a valid row address is asserted on the MA lines.
ROMCS2–ROMCS0	O	**ROM Chip Selects** are active Low outputs that provide the chip select for the BIOS ROM and/or the ROM/Flash memory array. After power-on reset, the ROMCS0 chip select will go active for accesses into the 64-Kbyte segment that contains the boot vector, at address 3FF0000h to 3FFFFFFh. ROMCS0 can be driven active during a linear (direct) address decode of certain addresses in the high memory (00A0000h–00FFFFFh) region. By default, direct-mapped accesses to the 64-Kbyte region from 00FFFF0h to 00FFFFFh are enabled to support Legacy PC/AT BIOS. This area is known as the aliased boot vector. It can also be activated by accessing a Memory Management System (MMS) page that points to the ROM0 address space. ROMCS1 is activated only when accessing an MMS page that points to it. A third, MMS-mappable ROMCS2 signal is available by reconfiguring one of the chip's General Purpose Input Output (GPIO) pins for this function and also requires the use of MMS to access devices connected to it.
ROMRD	O	**ROM Read** indicates that the current cycle is a read of the currently selected ROM device. When this signal is asserted, the selected ROM device can drive data onto the data bus.
ROMWR	O	**ROM Write** indicates that the current cycle is a write of the currently selected ROM device. When this signal is asserted, the selected ROM device can latch data from the data bus.
VL-Bus Interface		
VL_ADS	O	**Local Bus Address Strobe** is asserted to indicate the start of a VL-bus cycle. It is always strobed Low for one clock period. The address and status lines are valid on the rising edge of VL_LCLK, which samples this signal Low.
VL_BE3–VL_BE0	O	**Local Bus Byte Enables** indicate which byte lanes of the 32-bit data bus are involved with the current VL-bus transfer.
VL_BLAST	O	**Local Bus Burst Last** is asserted to indicate that the next VL_BRDY assertion will terminate the current VL-bus transfer.

Table 19. Signal Description Table (Continued)

Signal	Type	Description
VL_BRDY	I	**Local Bus Burst Ready** is asserted by the VL-bus target to indicate that it is terminating the current burst transfer. The chip samples this signal on the rising edge of VL_LCLK. VL_BRDY should be asserted for one VL_LCLK period per burst transfer. If VL_LRDY is asserted at the same time as VL_BRDY, VL_BRDY is ignored and the VL-bus transfer is terminated.
VL_D/C VL_M/IO VL_W/R	O O O	**Local Bus Data/Code Status** is driven Low to indicate that code is being transferred. A High on this signal indicates that data is being transferred. **Local Bus Memory/I/O Status** is driven Low to indicate an I/O transfer. A High on this signal indicates a memory transfer. **Local Bus Write/Read Status** is driven Low to indicate a read transfer. A High on this signal indicates a write.
VL_LCLK	O	**Local Bus Clock** is the VL-bus clock. It is used by the VL-bus target for all timing references. This signal is in phase with the internal CPU's clock input. (Rising Edge Active)
VL_LDEV	I	**Local Bus Device Select** is asserted by the VL-bus target to indicate that it is accepting the current transfer as indicated by the address and status lines. The VL-bus target asserts this signal as a function of the address and status presented on the bus.
VL_LRDY	I	**Local Bus Ready** is asserted by the VL-bus target to indicate that it is terminating the current bus cycle. This signal is sampled by the chip on the rising edge of VL_LCLK.
VL_RST	O	**Local Bus Reset** is the VL-bus master reset. It is controlled with CSC index 14h[4].
Power Management		
ACIN	I	**AC Supply Active** indicates to the system that it is being powered from an AC source. When asserted, this signal can disable power management functions (if configured to do so).
BL2–BL0	I	**Battery Low Detects** indicate to the chip the current status of the system's primary battery pack. BL0–BL2 can indicate various conditions of the battery as conditions change. These inputs can be used to force the system into one of the power saving modes when activated (Low-going Edge).
LBL2	O	**Latched Battery Low Detect 2** can be driven Low and latched on the low-going edge of the BL2 input to indicate to the system that the chip has been forced into the Suspend mode by a battery dead indication from the BL2 signal. It is cleared by one of the "all clear" indicators that allow the system to resume after a battery dead indication.
SUS_RES	I	**Suspend/Resume Operation:** When the chip is in Hyper-Speed, High-Speed, Low-Speed, or Standby mode, a software-configurable edge on this pin can cause the internal logic to enter Suspend mode. When in Suspend, a software-configurable edge on this pin can cause the chip to enter the High-Speed or Low-Speed mode. The choice of edge is configured using the SUS_RES Pin Configuration Register at CSC index 50h.

Bus Cycle Initiated	VL_M/IO	VL_D/C	VL_W/R
Interrupt Acknowledge	0	0	0
Halt/Special Cycle	0	0	1
I/O Read	0	1	0
I/O Write	0	1	1
Code Read	1	0	0
Reserved	1	0	1
Memory Read	1	1	0
Memory Write	1	1	1

Table 19. Signal Description Table (Continued)

Signal	Type	Description
Clocks		
32KXTAL1 32KXTAL2		**32.768-kHz Crystal Interface Signals** are used for the 32.768-kHz crystal. This is the main clock source for the chip and drives the internal Phase-Locked Loops (PLLs) that generate all other clock frequencies needed in the system.
CLK_IO	I/O	**Clock Input/Output** is an input to drive the integrated 8254 timer with a 1.19318-MHz clock signal from an external source, or an output to bring out certain internal clock sources to drive external devices.
LF_INT, LF_LS, LF_VID, LF_HS	A	**Loop Filters** connect external RC loop filters required by the internal PLLs. LF_VID is not supported on the ÉlanSC410 microcontroller.
Parallel Port *(Note: The names in parentheses in this section are those used in EPP mode.)*		
$\overline{\text{ACK}}$ (INTR)	I	**Printer Acknowledge:** In standard mode, this signal is driven by the parallel port device with the state of the printer acknowledge signal. In EPP mode, this signal indicates to the chip that the parallel port device has generated an interrupt request.
$\overline{\text{AFDT}}$ ($\overline{\text{DSTRB}}$)	O	**Auto Line Feed Detect:** In standard mode, this signal is driven by the chip indicating to the parallel port device to insert a line feed at the end of every line. In EPP mode, this signal is driven active by the chip during reads or writes to the EPP data registers.
BUSY ($\overline{\text{WAIT}}$)	I	**Printer Busy:** In standard mode, this signal is driven by the parallel port device with the state of the printer busy signal. In EPP mode, this signal adds wait states to the current cycle.
$\overline{\text{ERROR}}$	I	**Error:** The printer asserts this signal to inform the parallel port of a deselect condition, paper end (PE) or other error condition.
$\overline{\text{INIT}}$	O	**Initialize Printer:** This signals the printer to begin an initialization routine.
PE	I	**Paper End:** The printer asserts this signal when it is out of paper.
$\overline{\text{PPDWE}}$	O	**Parallel Port Write Enable** controls an external 374 type latch in a unidirectional parallel port design. This device latches the SD7–SD0 bus onto the parallel port data bus. To implement a bidirectional parallel port, this pin can be reconfigured to act as an address decode for the parallel port data port. $\overline{\text{PPDWE}}$ can then be externally gated with $\overline{\text{IOR}}$ and $\overline{\text{IOW}}$ to provide the Parallel Port Data Read and Write Strobes, respectively.
$\overline{\text{PPOEN}}$	O	**Parallel Port Output Buffer Enable** supports a bidirectional parallel port design. $\overline{\text{PPOEN}}$ controls the output enable of the external Parallel Port Output Buffer (373 octal D-type transparent latch).
SLCT	I	**Printer Select** is returned by a printer upon receipt of $\overline{\text{SLCTIN}}$.
$\overline{\text{SLCTIN}}$ ($\overline{\text{ASTRB}}$)	O	**Printer Selected:** In Standard mode, this signal is driven by the chip to select the parallel port device. In EPP mode, this signal is driven active by the chip during reads or writes to the EPP address register.
$\overline{\text{STRB}}$ ($\overline{\text{WRITE}}$)	O	**Strobe:** In Standard mode, this signal indicates to the parallel port device to latch the data on the parallel port data bus. In EPP mode, this signal is driven active during writes to the EPP data or the EPP address register.
Serial Port		
$\overline{\text{CTS}}$	I	**Clear To Send** is driven back to the serial port to indicate that the external data carrier equipment (DCE) is ready to accept data.
$\overline{\text{DCD}}$	I	**Data Carrier Detect** is driven back to the serial port from a piece of data carrier equipment when it has detected a carrier signal from a communications target.
$\overline{\text{DSR}}$	I	**Data Set Ready** indicates that the external DCE is ready to establish a communication link with the internal serial port controller.
$\overline{\text{DTR}}$	O	**Data Terminal Ready** indicates to the external DCE that the internal serial port controller is ready to communicate.
$\overline{\text{RIN}}$	I	**Ring Indicate** is used by an external modem to inform the serial port that a ring signal was detected. A change in state on this signal by the external modem can be configured to cause a modem status interrupt. This signal can be used to cause the chip to resume from a Suspend state.

Table 19. Signal Description Table (Continued)

Signal	Type	Description
RTS	O	**Request To Send** indicates to the external DCE that the internal serial port controller is ready to send data.
SIN	I	**Serial Data In** receives the serial data from the external serial device or DCE into the internal serial port controller.
SOUT	O	**Serial Data Out** transmits the serial data from the internal serial port controller to the external serial device or DCE.
Keyboard Interfaces		
KBD_COL7–KBD_COL0	O	**Matrix-Scanned Keyboard Column Outputs** drive the matrix keyboard column lines. (Open Collector Output with programmable termination)
KBD_ROW14–KBD_ROW0	STI	**Matrix-Scanned Keyboard Row Inputs** samples the row lines on the matrix keyboard.
XT_CLK	I/O	**XT Keyboard Clock** is the clock signal for an external XT keyboard interface. (Open Collector Output)
XT_DATA	I/O	**XT Keyboard Data** is the data signal for an external XT keyboard interface. (Open Collector Output)
General-Purpose Input/Output		
GPIO31–GPIO15 GPIO_CS14–GPIO_CS0	B	**General Purpose I/Os and Programmable Chip Selects** Each of the GPIOs can be programmed to be an input or an output. As outputs, all of the GPIOs can be programmed to be High or Low. Some of the GPIOs can be programmed to be High or Low for each of the power management modes. Also as outputs, some of these pins can be individually programmed as chip selects for other external peripheral devices. These can be configured as direct memory address decodes or I/O decodes qualified or non-qualified by the ISA bus command signals. Any one of the GPIO_CSx signals can be configured as ROMCS2. As inputs, all the GPIOs can be read back with a register bit. Some of these pins can be individually programmed to act as activity triggers, wake-up sources, or SMIs.
Serial Infrared Port		
SIRIN	I	**Infrared Serial Input** is the digital input for the serial infrared interface.
SIROUT	O	**Infrared Serial Output** is the digital output for the serial infrared interface.
PC Card Controller (ÉlanSC400 Microcontroller Only) ***(Note: The names in parentheses in this section are those used in PC Card Memory and I/O mode.)***		
BVD1_A (STSCHG_A)–BVD1_B (STSCHG_B)	I	**Battery Voltage Detect** is driven Low by a PC Card when its on-board battery is dead. When the PC Card interface is configured for I/O, this signal can be driven by the card to indicate a card status change. It is typically used to generate a system IRQ in this mode. These signals are not supported on the ÉlanSC410 microcontroller.
BVD2_A (SPKR_A) (DRQ_A)–BVD2_B (SPKR_B) (DRQ_B)	I	**Battery Voltage Detect** is driven Low by a PC Card when its on-board battery is weak. When the PC Card interface is configured for I/O, this signal can be driven by the card's speaker output. When enabled, this signal can drive the chip SPKR output. When PC Card DMA is enabled, the DMA request from the PC Card can be programmed to appear on this signal. See also the description for WP_A (IOIS16_A) (DRQ_A) and WP_B (IOIS16_B) (DRQ_B); the DMA request can also be programmed to appear on these pins. These signals are not supported on the ÉlanSC410 microcontroller.
CD_A–CD_B CD_A2	I	**Card Detect** indicates that the card is properly inserted. Socket A is capable of being configured to use two card detect inputs and Socket B is only provided with one. If only one card detect is to be used for a socket, the input signals should be driven from a logical AND (digital OR) of the CD1 and CD2 signals from their respective card interfaces. These signals are not supported on the ÉlanSC410 microcontroller.

Table 19. Signal Description Table (Continued)

Signal	Type	Description
ICDIR	O	**Card Data Direction** controls the direction of the card data buffers or voltage translators. It works with the MCEL and MCEH card enable signals to control data buffers on the card interface. When this signal is High, the data flow is from the chip to the card socket, indicating a data write cycle. When this signal is Low, the data flow is from the card socket into the chip, indicating a read cycle. This signal is not supported on the ÉlanSC410 microcontroller.
MCEH_A, MCEH_B	O	**Card Enables, High Byte** enables a PC Card's high data bus byte transceivers for the respective card interfaces. These signals are not supported on the ÉlanSC410 microcontroller.
MCEL_A, MCEL_B	O	**Card Enables, Low Byte** enables a PC Card's low data bus byte transceivers for the respective card interfaces. These signals are not supported on the ÉlanSC410 microcontroller.
OE	O	**PC Card Output Enable:** This is the PC Card memory read signal. This signal is not supported on the ÉlanSC410 microcontroller.
PCMA_VCC	O	**PC Card Socket A V_{CC} Enable** can be used to control the V_{CC} to Socket A. This signal is not supported on the ÉlanSC410 microcontroller.
PCMA_VPP2–PCMA_VPP1	O	**PC Card Socket A VPP Selects** can be used to control the V_{PP} to Socket A. These signals are not supported on the ÉlanSC410 microcontroller.
PCMB_VCC	O	**PC Card Socket B V_{CC} Enable** can be used to control the V_{CC} to Socket B. This signal is not supported on the ÉlanSC410 microcontroller.
PCMB_VPP2–PCMB_VPP1	O	**PC Card Socket B VPP Selects** can be used to control the V_{PP} to Socket B. These signals are not supported on the ÉlanSC410 microcontroller.
RDY_A (IREQ_A), RDY_B (IREQ_B)	I	**Card Ready** indicates that the respective card is ready to accept a new data transfer command. When the card interface is configured as an I/O interface, this signal is used as the card Interrupt Request input into the chip. These signals are not supported on the ÉlanSC410 microcontroller.
REG_A (DACK_A), REG_B (DACK_B)	O	**Attribute Memory Select** signals are driven inactive (High) for accesses to a PC Card's common memory, and asserted (Low) for accesses to a PC Card's attribute memory and I/O space for their respective card interfaces. When PC Card DMA is enabled, the DMA acknowledge to the PC Card appears on this signal. These signals are not supported on the ÉlanSC410 microcontroller.
RST_A, RST_B	O	**Card Reset** signals are the reset for their respective cards. When active, this signal clears the Interrupt and General Control Register (PC Card index 03h and 43h), thus placing a card in an unconfigured (Memory-Only mode) state. It also indicates the beginning of any additional card initialization. These signals are not supported on the ÉlanSC410 microcontroller.
WAIT_AB	I	**Extend Bus Cycle** delays the completion of the memory access or I/O access that is currently in progress. When this signal is asserted (Low), wait states are inserted into the cycle in progress. Only one WAIT input is provided on the chip. External logic is required for a two-socket implementation to logically AND (digitally OR) each card's WAIT signal together. This signal is not supported on the ÉlanSC410 microcontroller.
WE (TC)	O	**PC Card Write Enable** is the PC Card memory write signal. Data is transferred from the chip to the PC Card. When PC Card DMA is enabled, the DMA Terminal Count to the PC Card appears on this signal. This signal is not supported on the ÉlanSC410 microcontroller.
WP_A (IOIS16_A) (DRQ_A), WP_B (IOIS16_B) (DRQ_B)	I	**Write Protect** indicates the status of the respective card's Write Protect switch. When the respective card is configured for an I/O interface, this signal is used by the card to indicate back to the chip that the currently accessed port is 16 bits wide. When PC Card DMA is enabled, the DMA request from the PC Card can be programmed to appear on this signal. See also the description for BVD2_A (SPKR_A) (DRQ_A) and BVD2_B (SPKR_B) (DRQ_B); the DMA request can also be programmed to appear on these pins. These signals are not supported on the ÉlanSC410 microcontroller.
LCD Graphics Controller (ÉlanSC400 Microcontroller Only)		
FRM	O	**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). This signal is not supported on the ÉlanSC410 microcontroller.

Table 19. Signal Description Table (Continued)

Signal	Type	Description
LC	O	**LCD Panel Line Clock** is activated at the start of every pixel line. It is commonly referred to by LCD data sheets as CL1 or CP1. This signal is not supported on the ÉlanSC410 microcontroller.
LCDD7–LCDD0	O	**LCD Panel Data bits:** LCDD7–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. In dual-scan panel mode, LCDD3–LCDD0 are the data bits for the top half of the LCD, and LCDD7–LCDD4 are the data bits for the bottom half of the LCD. When driving 8-bit single-scan panels (monochrome or color STN), these bits are the 8-bit data interface. These signals are not supported on the ÉlanSC410 microcontroller.
$\overline{\text{LVDD}}$	O	**LCD Panel VDD Voltage Control** is used to control the assertion of the LCD's V_{DD} voltage. This is provided to be part of the solution in sequencing the panel's V_{DD}, DATA, and V_{EE} in the proper order during panel power-up and power-down to prevent damage to the panel from CMOS driver latch up. V_{DD} is used to power the LCD logic and is usually +5 V or +3 V DC. This signal is not supported on the ÉlanSC410 microcontroller.
$\overline{\text{LVEE}}$	O	**LCD Panel VEE Voltage Control** is used to control the assertion of the LCD's V_{EE} voltage. This is provided to be part of the solution in sequencing the panel's V_{DD}, DATA, and V_{EE} in the proper order during panel power-up and power-down to prevent damage to the panel from CMOS driver latch up. V_{EE} is the LCD contrast voltage and is either positive or negative with an amplitude of 15–30 V DC.This signal is not supported on the ÉlanSC410 microcontroller.
M	O	**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. This signal is not supported on the ÉlanSC410 microcontroller.
SCK	O	**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. This signal is not supported on the ÉlanSC410 microcontroller.
Boundary Scan Test Interface		
BNDSCN_TCK	I	**Test Clock** is the boundary-scan input clock that is used to shift serial data patterns in from BNDSCN_TDI.
BNDSCN_TDI	I	**Test Data Input** is the serial input stream for boundary-scan input data. This pin has a weak internal pullup resistor. It is sampled on the rising edge of BNDSCN_TCK. If not driven, this input is sampled High internally.
BNDSCN_TDO	O TS	**Test Data Output** is the serial output stream for boundary-scan result data. It is in the high-impedance state except when scanning is in progress.
BNDSCN_TMS	I	**Test Mode Select** is an input for controlling the test access port. This pin has a weak internal pullup resistor. If it is not driven, it is sampled High internally.
Reset and Power		
BBATSEN	A	**Backup Battery Sense:** RTC (Real Time Clock) backup battery voltage is sampled on this pin each time the AV_{CC} pin has power applied to it followed by a chip master reset. If this samples below 2.4 V, the VRT bit (RTC index 0Dh) is cleared until read one time. At this time, the VRT bit is set until BBATSEN is sampled again. BBATSEN also provides a power-on-reset signal for the RTC when an RTC backup battery is applied for the first time.
GND		**Ground Pins**
$\overline{\text{RESET}}$	I	**Reset Input** is an asynchronous hardware reset input equivalent to POWERGOOD in the AT system architecture.
V_{CC}		**3.3-V DC Supply Pins** provide power to the discrete logic and I/O pins.
V_{CC}_A	Analog	**3.3-V Supply Pins** provide power to the analog section of the chip, including the internal PLLs and integrated oscillator circuit. Extreme care should be taken that this supply voltage is isolated properly to provide a clean, noise free voltage to the PLLs.
V_{CC}_CPU	CPU	**3.3-V DC Supply Pins** provide power to the internal CPU.
V_{CC}_RTC	RTC	**3.3-V Supply Pin** provides power to the internal real-time clock and on-board static/ configuration RAM. This pin can be driven independently of all other power pins.

Multiplexed Pin Function Options

Table 20 shows how to configure each multiplexed signal on the ÉlanSC400 and ÉlanSC410 microcontrollers.

Note that those signals marked with a superscript 1 ([1]) are not supported on the ÉlanSC410 microcontroller.

Pins with multiplexed functions have their functions selected in one of three ways:

- By configuration pins that are latched during reset
- By assertion at BNDSCN_EN
- By firmware via programmed configuration registers

Table 20. Multiplexed Pin Configuration Options

Signal You Want	Signals You Give Up	How to Configure the Signal You Want on the Pin	Pin #
System Interface			
BALE	KBD_ROW10	Set CSC index 39h[2].	D2
$\overline{\text{DBUFOE}}$	GPIO_CS4	Hardwire strap the CFG3 pin High.	C4
$\overline{\text{DBUFRDH}}$	GPIO_CS3	Hardwire strap the CFG3 pin High.	D17
$\overline{\text{DBUFRDL}}$	GPIO_CS2	Hardwire strap the CFG3 pin High.	C18
$\overline{\text{MCS16}}$	KBD_ROW12	Set CSC index 39h[2].	F3
$\overline{\text{PDACK1}}$	KBD_ROW7	Set CSC index 39h[2].	D3
PDRQ1	KBD_ROW8	Set CSC index 39h[2].	C2
PIRQ0	GPIO_CS8	Set CSC index 38h[1].	W18
PIRQ1	GPIO_CS7	Set CSC index 38h[2].	Y19
PIRQ2	KBD_ROW9	Set CSC index 39h[2].	E3
PIRQ3	KBD_COL2	Set CSC index 3Ah[1].	A2
PIRQ4	KBD_COL3	Set CSC index 3Ah[1].	B3
PIRQ5	KBD_COL4	Set CSC index 3Ah[2].	C3
PIRQ6	KBD_COL5	Set CSC index 3Ah[2].	A1
PIRQ7	KBD_COL6	Set CSC index 3Ah[2].	B2
$\overline{\text{R32BFOE}}$	KBD_ROW13	Hardwire-strap both the $\overline{\text{CFG1}}$ and CFG0 pins High to enable the $\overline{\text{32-bit ROM}}$ interface on $\overline{\text{ROMSC0}}$. This automatically enables $\overline{\text{R32BFOE}}$.	A3
$\overline{\text{SBHE}}$	KBD_ROW11	Set CSC index 39h[2].	C1
Configuration Pins (Pinstraps) (See “Using the Configuration Pins to Select Pin Functions” on page 74.)			
Memory Interface			
$\overline{\text{CASH2}}$	KBD_ROW2	Set bit 3 of the DRAM Bank x Configuration Register. Set CSC index 00h[7] and 00h[3], or Set CSC index 01h[7] and 01h[3], or Set CSC index 02h[7], or Set CSC index 03h[7].	C17
$\overline{\text{CASH3}}$	KBD_ROW3		B18
$\overline{\text{CASL2}}$	KBD_ROW0		B19
$\overline{\text{CASL3}}$	KBD_ROW1		D16
MA12	KBD_ROW6		B17
$\overline{\text{RAS2}}$	KBD_ROW4		D15
$\overline{\text{RAS3}}$	KBD_ROW5		C16

Table 20. Multiplexed Pin Configuration Options (Continued)

Signal You Want	Signals You Give Up	How to Configure the Signal You Want on the Pin	Pin #
VL-Bus Interface			
VL_ADS	LCDD1[1]	Enable the VL-bus interface by setting CSC index 14h[3].	C19
VL_BE0	SCK[1]		F19
VL_BE1	LC[1]		E20
VL_BE2	M[1]		F18
VL_BE3	LCDD7[1]		D20
VL_BLAST	LVDD[1]		A19
VL_BRDY	LVEE[1]		A20
VL_D/C	LCDD5[1]		E18
VL_LCLK	FRM[1]		E19
VL_LDEV	LCDD6[1]		F17
VL_LRDY	LCDD4[1]		D19
VL_M/IO	LCDD3[1]		C20
VL_RST	LCDD0[1]		B20
VL_W/R	LCDD2[1]		D18
ISA Bus			
AEN	GPIO_CS10	Set CSC index 38h[0].	V17
IOCHRDY	GPIO_CS6	Set CSC index 38h[3].	V18
IOCS16	GPIO_CS5	Set CSC index 38h[4].	W19
PDACK0	GPIO_CS11	Set CSC index 38h[0]	W17
PDRQ0	GPIO_CS12	Set CSC index 38h[0].	Y17
TC	GPIO_CS9	Set CSC index 38h[0].	Y18
GPIOs			
GPIO15	PCMA_VPP2[1]	Clear CSC index 39h[5].	Y15
GPIO16	PCMB_VCC[1]	Clear CSC index 39h[6].	V15
GPIO17	PCMB_VPP1[1]	Clear CSC index 39h[6].	W15
GPIO18	PCMB_VPP2[1]	Clear CSC index 39h[6].	Y14
GPIO19	LBL2	Clear CSC index 39h[4].	V14
GPIO20	CD_A2[1]	Clear CSC index 3Ah[0].	G18
GPIO21	PPDWE	Clear CSC index 39h[1–0].	V3
GPIO22	PPOEN	Clear CSC index 39h[1–0].	T3
GPIO23	SLCT, WP_B[1]	Clear CSC index 39h[1–0].	U4
GPIO24	BUSY, BVD2_B[1]	Clear CSC index 39h[1–0].	U3
GPIO25	ACK, BVD1_B[1]	Clear CSC index 39h[1–0].	U2
GPIO26	PE, RDY_B[1]	Clear CSC index 39h[1–0].	W2
GPIO27	ERROR, CD_B[1]	Clear CSC index 39h[1–0].	V4
GPIO28	INIT, REG_B[1]	Clear CSC index 39h[1–0].	Y2
GPIO29	SLCTIN, RST_B[1]	Clear CSC index 39h[1–0].	W3
GPIO30	AFDT, MCEH_B[1]	Clear CSC index 39h[1–0].	W1
GPIO31	STRB, MCEL_B[1]	Clear CSC index 39h[1–0].	V2
GPIO_CS2	DBUFRDL	Hardwire-strap the CFG3 pin Low.	C18
GPIO_CS3	DBUFRDH	Hardwire-strap the CFG3 pin Low.	D17
GPIO_CS4	DBUFOE	Hardwire-strap the CFG3 pin Low.	C4
GPIO_CS5	IOCS16	Clear CSC index 38h[4].	W19
GPIO_CS6	IOCHRDY	Clear CSC index 38h[3].	V18
GPIO_CS7	PIRQ1	Clear CSC index 38h[2].	Y19
GPIO_CS8	PIRQ0	Clear CSC index 38h[1].	W18
GPIO_CS9	TC	Clear CSC index 38h[0].	Y18
GPIO_CS10	AEN	Clear CSC index 38h[0].	V17

Table 20. Multiplexed Pin Configuration Options (Continued)

Signal You Want	Signals You Give Up	How to Configure the Signal You Want on the Pin	Pin #
GPIO_CS11	PDACK0	Clear CSC index 38h[0].	W17
GPIO_CS12	PDRQ0	Clear CSC index 38h[0].	Y17
GPIO_CS13	PCMA_VCC[1]	Clear CSC index 39h[5].	V16
GPIO_CS14	PCMA_VPP1[1]	Clear CSC index 39h[5].	W16
Parallel Port			
ACK	GPIO25, BVD1_B[1]	Write CSC index 39h[1–0] to 10.	U2
AFDT	GPIO30, MCEH_B[1]		W1
BUSY	GPIO24, BVD2_B[1]		U3
ERROR	GPIO27, CD_B[1]		V4
INIT	GPIO28, REG_B[1]		Y2
PE	GPIO26, RDY_B[1]		W2
PPDWE	GPIO21		V3
PPOEN	GPIO22		T3
SLCT	GPIO23, WP_B[1]		U4
SLCTIN	GPIO29, RST_B[1]		W3
STRB	GPIO31, MCEL_B[1]		V2
Keyboard Interface			
KBD_COL0	XT_DATA	Clear CSC index 39h[3].	E2
KBD_COL1	XT_CLK	Clear CSC index 39h[3].	D1
KBD_COL2	PIRQ3	Clear CSC index 3Ah[1].	A2
KBD_COL3	PIRQ4	Clear CSC index 3Ah[1].	B3
KBD_COL4	PIRQ5	Clear CSC index 3Ah[1].	C3
KBD_COL5	PIRQ6	Clear CSC index 3Ah[1].	A1
KBD_COL6	PIRQ7	Clear CSC index 3Ah[1].	B2
KBD_ROW0	CASL2	Clear CSC index 00h[7] and 00h[3], or clear CSC index 01h[7] and 01h[3], or clear CSC index 02h[7], or clear CSC index 03h[7].	B19
KBD_ROW1	CASL3		D16
KBD_ROW2	CASH2		C17
KBD_ROW3	CASH3		B18
KBD_ROW4	RAS2		D15
KBD_ROW5	RAS3		C16
KBD_ROW6	MA12		B17
KBD_ROW7	PDACK1	Clear CSC index 39h[2].	D3
KBD_ROW8	PDRQ1	Clear CSC index 39h[2].	C2
KBD_ROW9	PIRQ2	Clear CSC index 39h[2].	E3
KBD_ROW10	BALE	Clear CSC index 39h[2].	D2
KBD_ROW11	SBHE	Clear CSC index 39h[2].	C1
KBD_ROW12	MCS16	Clear CSC index 39h[2].	F3
KBD_ROW13	R32BFOE	Do not enable the 32-bit ROM interface on ROMCS0 (e.g., do not hardwire-strap both the CFG1 and CFG0 pins High).	A3
XT_CLK	KBD_COL1	Clear CSC index 39h[3].	D1
XT_DATA	KBD_COL0	Clear CSC index 39h[3].	E2
PC Card (ÉlanSC400 Microcontroller Only)			
BVD1_B[1]	GPIO25, ACK	Write CSC index 39h[1–0] to 01.	U2
BVD2_B[1]	GPIO24, BUSY	Write CSC index 39h[1–0] to 01.	U3
CD_A2[1]	GPIO20	Set CSC index 3Ah[0].	G18
CD_B[1]	GPIO27, ERROR	Write CSC index 39h[1–0] to 01.	V4
LBL2[1]	GPIO19	Set CSC index 39h[4].	V14
MCEH_A[1]	BNDSCN_TMS	Pull the BNDSCN_EN pin Low.	N3
MCEH_B[1]	GPIO30, AFDT	Write CSC index 39h[1–0] to 01.	W1
MCEL_A[1]	BDNSCN_TCK	Pull the BNDSCN_EN pin Low.	P2

Table 20. Multiplexed Pin Configuration Options (Continued)

Signal You Want	Signals You Give Up	How to Configure the Signal You Want on the Pin	Pin #
MCEL_B[1]	GPIO31, STRB	Write CSC index 39h[1–0] to 01.	V2
PCMA_VCC[1]	GPIO_CS13	Set CSC index 39h[5].	V16
PCMA_VPP1[1]	GPIO_CS14	Set CSC index 39h[5].	W16
PCMA_VPP2[1]	GPIO15	Set CSC index 39h[5].	Y15
PCMB_VCC[1]	GPIO16	Set CSC index 39h[6].	V15
PCMB_VPP1[1]	GPIO17	Set CSC index 39h[6].	W15
PCMB_VPP2[1]	GPIO18	Set CSC index 39h[6].	Y14
RDY_B[1]	GPIO26, PE	Write CSC index 39h[1–0] to 01.	W2
REG_A[1]	BNDSCN_TDO	Pull the BNDSCN_EN pin Low.	M2
REG_B[1]	GPIO28, INIT	Write CSC index 39h[1–0] to 01.	Y2
RST_A[1]	BNDSCN_TDI	Pull the BNDSCN_EN pin Low.	R1
RST_B[1]	GPIO29, SLCTIN	Write CSC index 39h[1–0] to 01.	W3
WP_B[1]	GPIO23, SLCT	Write CSC index 39h[1–0] to 01.	U4
LCD Graphics Controller (ÉlanSC400 Microcontroller Only)			
FRM[1]	VL_LCLK	Enable the graphics controller by setting CSC index DDh[2].	E19
LC[1]	VL_BE1		E20
LCDD0[1]	VL_RST		B20
LCDD1[1]	VL_ADS		C19
LCDD2[1]	VL_W/R		D18
LCDD3[1]	VL_M/IO		C20
LCDD4[1]	VL_LRDY		D19
LCDD5[1]	VL_D/C		E18
LCDD6[1]	VL_LDEV		F17
LCDD7[1]	VL_BE3		D20
LVDD[1]	VL_BLAST		A19
LVEE[1]	VL_BRDY		A20
M[1]	VL_BE2		F18
SCK[1]	VL_BE0		F19
Boundary Scan Interface			
BDNSCN_TCK	MCEL_A [1]	Pull the BNDSCN_EN signal High.	P2
BNDSCN_TDI	RST_A[1]		R1
BNDSCN_TDO	REG_A[1]		M2
BNDSCN_TMS	MCEH_A [1]		N3
Miscellaneous			
BL0	CLK_IO	Write CSC index 38h[7–6] to 01.	W14
CLK_IO	BL0	Write CSC index 38h[7–6] to 10 to enable CLK_IO as an output or to 11 to enable CLK_IO as a timer clock input.	W14

Notes:

1. This signal is not supported on the ÉlanSC410 microcontroller.

Using the Configuration Pins to Select Pin Functions

The configuration pins are used only for those functions that must be selected at reset, prior to firmware execution. All other I/O functions are selected using configuration registers.

Table 21 provides an overview of the configuration pin functions. All of the CFG pins have weak internal pull-down resistors that select the default function. External pullup resistors are required to select an alternative function.

Table 21. Pinstrap Bus Buffer Options

CFG3 (1)	CFG1	CFG0	ROMCS0 Data Width	DBUFOE DBUFRDL DBUFRDH	R32BFOE
0	0	0	x8	Disabled	Disabled
0	0	1	Reserved	Reserved	Reserved
0	1	0	x16	Disabled	Disabled
0	1	1	x32[2]	Disabled	Enabled
1	0	0	x8	Enabled	Disabled
1	0	1	Reserved	Reserved	Reserved
1	1	0	x16	Enabled	Disabled
1	1	1	x32[2]	Enabled	Enabled

Notes:

1. *CFG3 is defined as the enable/disable for the DBUFOE, DBUFRDL, and DBUFRDH signals. They can be enabled independently of whether a x32 D bus is selected via the firmware to support the VL local bus or x32 DRAM interface.*
2. *The x32 ROM option must be selected for ROMCS0 for the R32BFOE signal to be enabled. The selection of the DBUFOE, DBUFRDL, and DBUFRDH signals are still dependent only on the CFG3 signal.*

CFG0 and CFG1 Pins

These pins (shown in Table 22) configure the data bus width (x8, x16, or x32) of the ROM interface that is selected by the ROMCS0 pin. If a x32 ROM is selected, these pins also enable the ROM x32 Data Bus Buffer Output Enable signal (R32BFOE). If a 32-bit data bus width is selected for the ROM interface, the R32BFOE signal will be asserted for all ROMCSx accesses to 32-bit ROM. Exercise caution because the data bus width for the ROMCS0 interface can also be changed through programming. This feature was implemented mainly for testing.

Table 22. CFG0 and CFG1 Configuration

CFG1	CFG0	Configuration
0	0	x8 ROMCS0 ROM interface
0	1	Reserved
1	0	x16 ROMCS0 ROM interface
1	1	x32 ROMCS0 ROM interface

CFG2 Pin—ÉlanSC400 Microcontroller Only

This configuration pin (see Table 23) is used on the ÉlanSC400 microcontroller to select the ROMCS0 steering at system boot time. The boot ROM chip select (ROMCS0) can either be enabled to drive the ROMCS0 pin or can be rerouted to drive the PC Card (Socket A only) interface chip selects. The CFG0 and CFG1 pins are still used to select the data bus width for the ROMCS0 decode, regardless of the CFG2 configuration. The PC Card ROMCS0 redirection should not be selected when the CFG0 and CFG1 configuration pins are set to select a x32 ROM interface.

When the ROM chip select decode has been redirected to PC Card Socket A, all of the normal PC Card controller features can still be used to drive the PC Card Socket A interface. The ROM chip select decode remapping to the PC Card socket can be enabled and disabled using firmware at any time.

Table 23. CFG2 Configuration (ÉlanSC400 microcontroller only)

CFG2	Configuration
0	Enables the ROMCS0 decode on the ROMCS0 pin
1	Enables the ROMCS0 decode to access PC Card Socket A

CFG3 Pin

This configuration pin is used for selecting between the GPIO_CS4–GPIO_CS2 I/O pins and the SD bus buffer control signals: DBUFOE, DBUFRDL, and DBUFRDH. When the buffer control signal configuration is selected using the CFG3 pin, the DBUFOE, DBUFRDL, and DBUFRDH signals are driven from boot time on for all accesses to the peripheral data bus. These signals are used for the external system bus transceiver control. See Table 24 for the CFG3 configuration definitions.

Table 24. CFG3 Configuration

CFG3	Configuration
0	Enables the GPIO_CS4–GPIO_CS2 signals on the I/O pins
1	Enables the SD bus buffer control signals DBUFOE, DBUFRDL, and DBUFRDH on the I/O pins

BNDSCN_EN Pin

The BNDSCN_EN configuration pin (see Table 25) is used to enable the boundary scan function I/O pins. The following pins are configured for their boundary scan function when BNDSCN_EN is asserted:

- BNDSCN_TCK
- BNDSCN_TMS
- BNDSCN_TDI
- BNDSCN_TDO

Table 25. BNDSCN_EN Configuration

BNDSCN_EN	Configuration
0	Enables the PC Card function
1	Enables the boundary scan functions: BNDSCN_TCK, BNDSCN_TMS, BNDSCN_TDI, and BNDSCN_TDO

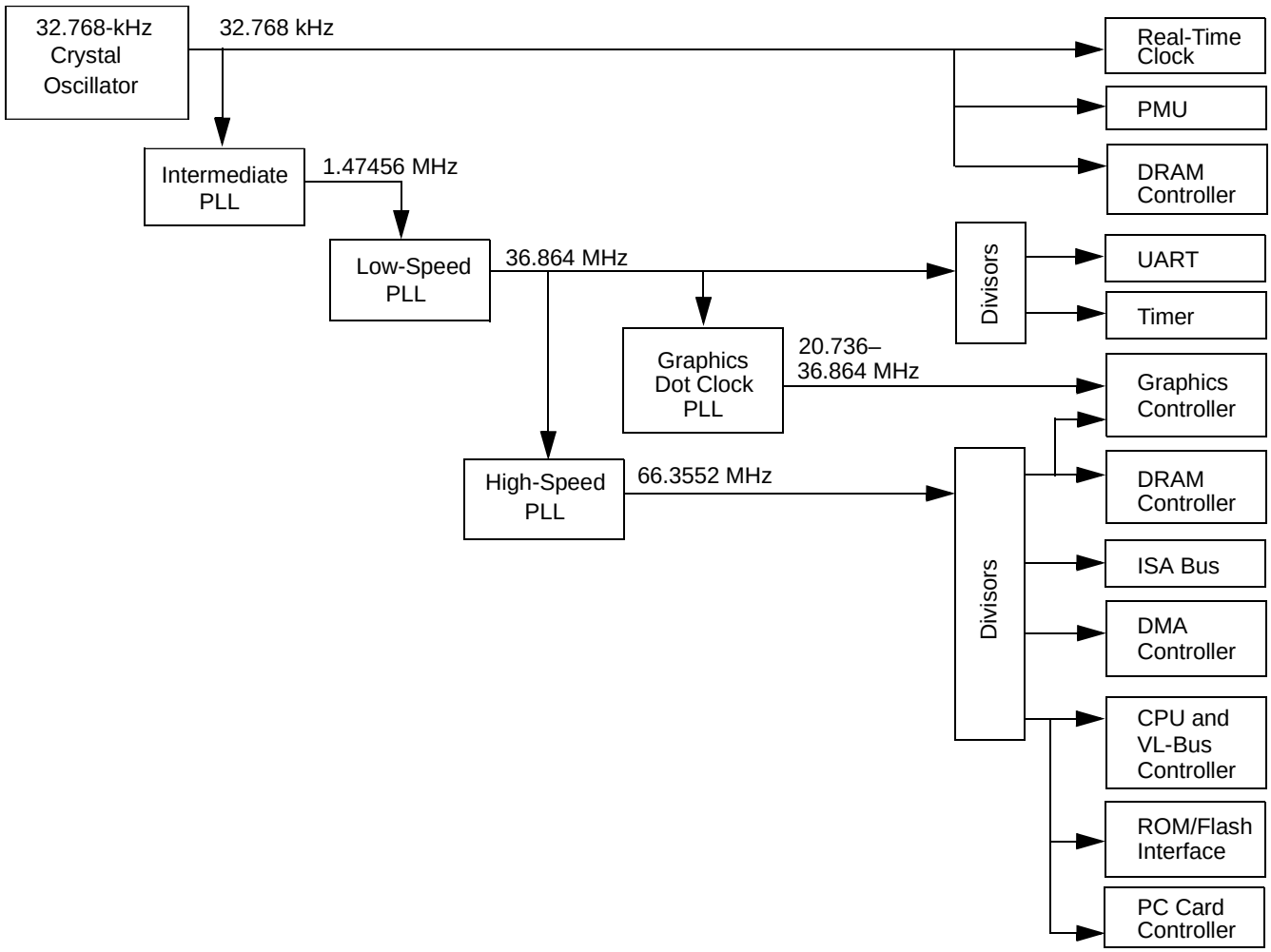
CLOCKING

Clock Generation

The ÉlanSC400 and ÉlanSC410 microcontrollers require only one 32.768-kHz crystal to generate all the other clock frequencies required by the system. The output of the on-chip crystal oscillator circuit is used to generate the various frequencies by utilizing four Phase-Locked Loop (PLL) circuits. The PLL clock distribution scheme is shown in Figure 4. Table 26 shows all the PLL output frequencies and their usage. (Note that these four PLL circuits are in addition to the internal CPU PLL and do not replace it.)

The crystal oscillator needs two pins, but it does not require any external components except the crystal; the load capacitors and the feedback resistor are integrated on-chip.

The four PLLs are called Intermediate PLL, Low-Speed PLL, High-Speed PLL, and Graphics Dot Clock PLL. Each of the integrated phase-locked loops has a dedicated pin to support the required external loop filter. These pins are: LF_INT (Intermediate PLL), LF_LS (Low-Speed PLL), LF_HS (High-Speed PLL), and LF_VID (Graphics Dot Clock PLL). (The LF_VID pin is not supported on the ÉlanSC410 microcontroller.) Two capacitors and one resistor are required to implement each loop filter.



Notes:

On the ÉlanSC400 microcontroller, the graphics controller's DRAM interface is clocked by the 66-MHz DRAM clock.

Both the ROM/Flash memory interface and the PC Card controller are clocked from the CPU clock. They also have the option to be run from the slow system clock.

Neither the graphics controller nor the PC Card controller are supported on the ÉlanSC410 microcontroller.

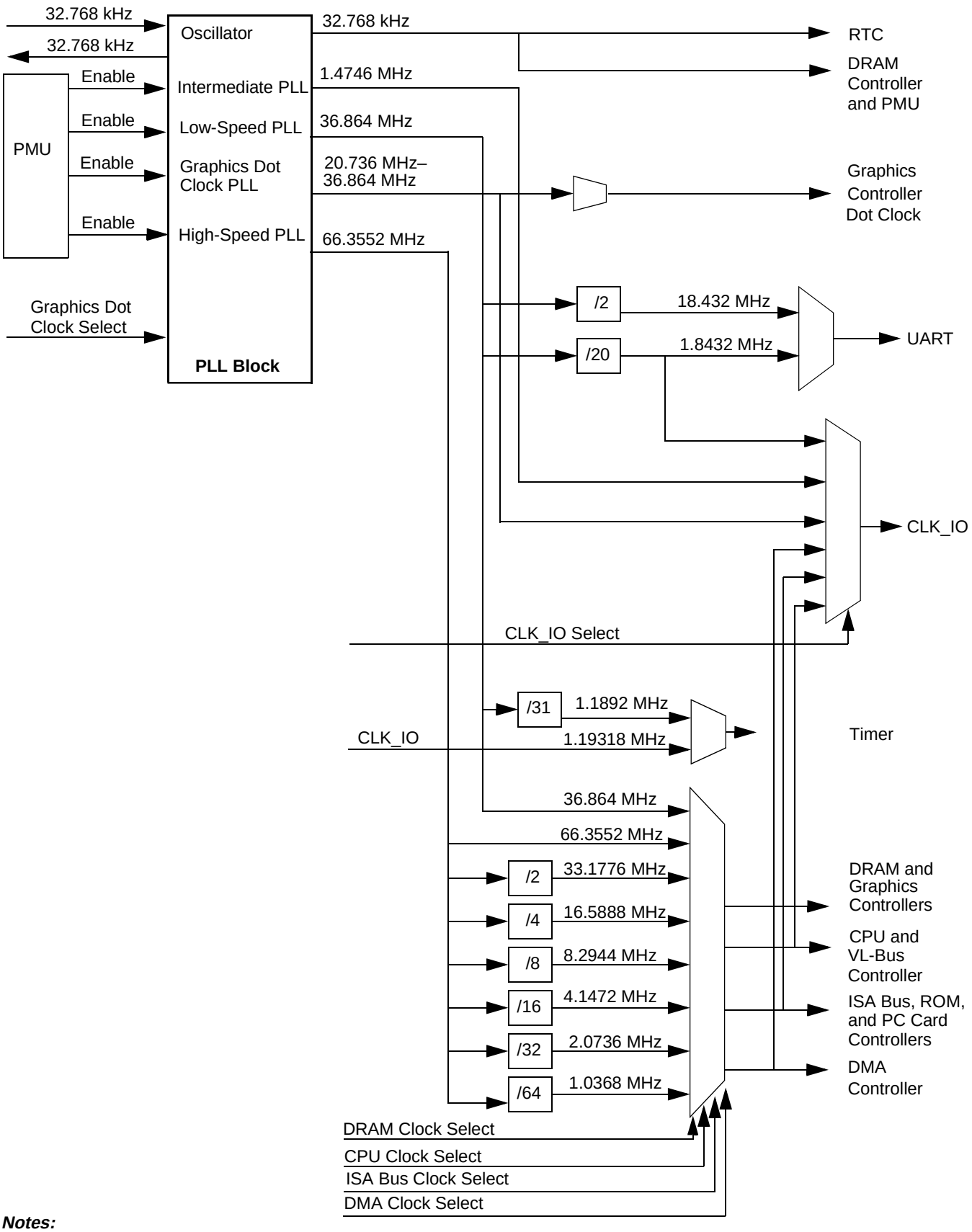
Figure 4. Clock Generation Block Diagram

Integrated Peripheral Clock Sources

Table 26 and Figure 5 show the primary peripheral clocks internal to the microcontroller and the PLL and divider sources that are used in the generation of these clocks. Note that several of the peripheral clocks are programmable. This programmability is either directly controlled by system firmware or is forced due to a power-management mode change. The graphics controller and the PC Card controller are not supported on the ÉlanSC410 microcontroller.

Table 26. Integrated Peripheral Clock Sources

Source PLL	Divider	Resulting Frequency	Where Used
Intermediate PLL 1.4746 MHz	1	1.4746 MHz	Low-speed PLL input
Low-speed PLL 36.864 MHz	1	36.864 MHz	High-speed PLL input Graphics dot clock PLL input
	20	1.8432 MHz	UART
	2	18.4328 MHz	UART
	31	1.1892 MHz	PIT
Graphics dot clock PLL 36.864 MHz	Programmable	20.736–36.864 MHz	Graphics controller dot clock
	1	36.864 MHz	Graphics controller
High-speed PLL 66.3552 MHz	1	66.3552 MHz	DRAM controller Graphics controller
	2	33.1776 MHz	CPU VL-bus controller
	4	16.5888 MHz	CPU VL-bus controller DMA controller
	8	8.2944 MHz	CPU VL-bus controller ISA bus controller ROM/Flash memory interface DMA controller PC Card controller
	16	4.1472 MHz	CPU VL-bus controller ISA bus controller ROM/Flash memory interface DMA controller PC Card controller
	32	2.0736 MHz	CPU VL-bus controller ISA bus controller ROM/Flash memory interface DMA controller PC Card controller
	64	1.0368 MHz	CPU VL-bus controller ISA bus controller ROM/Flash memory interface DMA controller PC Card controller



Notes:

The graphics controller and the PC Card controller are not supported on the ÉlanSC410 microcontroller.

Figure 5. Clock Source Block Diagram

32-kHz Crystal Oscillator

The 32-kHz oscillator circuit is shown in Figure 6; the only external component required for operation is a 32.768-kHz crystal. The inverting amplifier (AMP) is integrated on-chip together with the feedback resistor and the load capacitors. As shown in Figure 7, the on-chip oscillator circuit can be bypassed by removing the external crystal, grounding the 32KXTAL1 pin, and driving the 32KXTAL2 pin with an external 32-kHz clock. (The 32KXTAL2 pin should not exceed 2.0 V.) When 32KXTAL1 is grounded, the amplifier no longer affects the circuit.

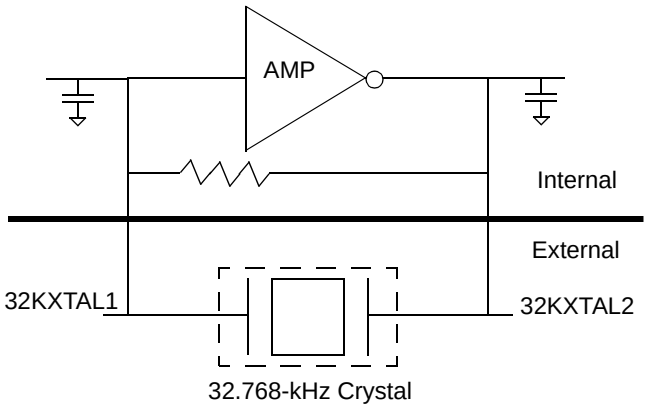


Figure 6. 32-kHz Crystal Circuit

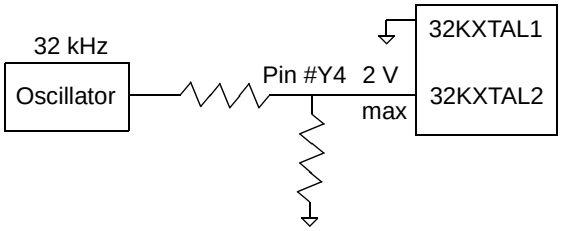


Figure 7. 32-kHz Oscillator Circuit

Loop Filters

Each of the PLLs in the ÉlanSC400 and ÉlanSC410 microcontrollers requires an external loop filter. For a cleaner circuit, the designer should consider the following:

- Place the loop filter components as close as possible to the loop filter signals (LF_INT, LF_LS, LF_HS, and LF_VID (ÉlanSC400 microcontroller only)), which are located in one corner of the microcontroller.
- Route the loop filter signals first and by hand.
- Keep all clocks and noisy signals away from the loop filter area (even on the inner layers).

For an even cleaner circuit, the designer could optionally place an analog V_{CC} power plane directly under the loop filter circuit.

The value of the loop filter parameters can also affect the performance of the filter. For example, the values of C1 and R affect lock time and jitter (increasing RC increases lock time and decreases jitter). The value of C2 can help clean up high-frequency noise. Note that using too large of values for the components can cause the PLL to become unstable. The loop filter component value specifications are shown in Table 28 on page 84.

Intermediate and Low-Speed PLLs

Figure 8 on page 80 shows the block diagram for both the Intermediate and Low-Speed PLLs. Each consists of a phase detector, a charge pump, a voltage controlled oscillator (VCO), an external loop filter, and a feedback divider. This is a generic implementation of the charge-pump PLL architecture; all four PLLs use the same architecture. The Intermediate and Low-Speed PLLs differ only in component values and frequency of operation.

The phase detector compares the phase and frequency of the two clock signals, reference frequency (Fr) and feedback frequency (Ff). The Up signal is a logic 1 if Fr leads Ff, while the Down signal is a logic 1 if Ff leads Fr. The Up and Down signals control the charge pump. The charge pump either charges or discharges the loop filter capacitors to change the VCO input voltage level. Because the VCO output frequency tracks the VCO input voltage, the VCO output frequency is adjusted whenever Fr and Ff differ in phase or frequency.

The feedback divide ratio determines the frequency multiplication factor.

Frequency multiplication is 1/(Feedback Divider).

For the Intermediate PLL, the feedback divider is 1/45; therefore, the frequency multiplication is 45. With an input frequency of 32.768 kHz, the output frequency is 1.47456 MHz.

The input clock for the Low-Speed PLL, Fr, originates at the Intermediate PLL output. Fr is multiplied by 25 to generate the 36.864-MHz clock output.

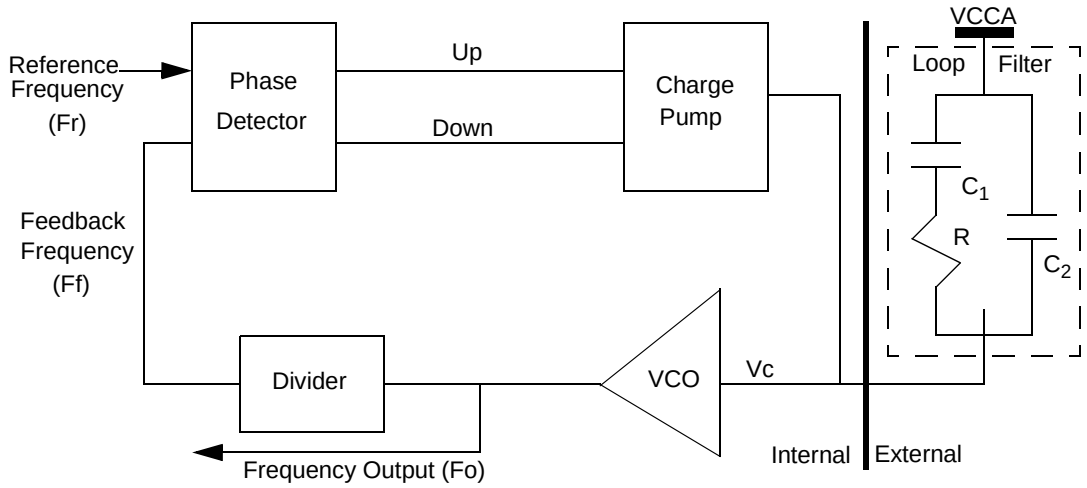


Figure 8. Intermediate and Low-Speed PLLs Block Diagram

Graphics Dot Clock PLL (ÉlanSC400 Microcontroller Only)

The input clock to the Graphics Dot Clock PLL is the output clock (36.864 MHz) of the Low-Speed PLL divided by 16. The Graphics Dot Clock PLL is not supported on the ÉlanSC410 microcontroller. The output frequency is programmable using three extended register bits (PLLRATIO[2–0]) in the range of 20.736 MHz to 36.864 MHz (spaced 2.304 MHz apart). These three bits (in graphics index register 4Ch) control the output frequency by selecting the divide value in the feedback divider as shown in Table 27.

The Graphics Dot Clock PLL requires a stabilization period after changing frequency. Figure 9 shows the block diagram for the Graphics Dot Clock PLL.

Table 27. Frequency Selection Control for Graphics Dot Clock PLL

PLLRATIO[2–0]	Divider	Output Frequency (MHz)
000	9	20.736
001	10	23.04
010	11	25.344
011	12	27.648
100	13	29.952
101	14	32.256
110	15	34.56
111	16	36.864

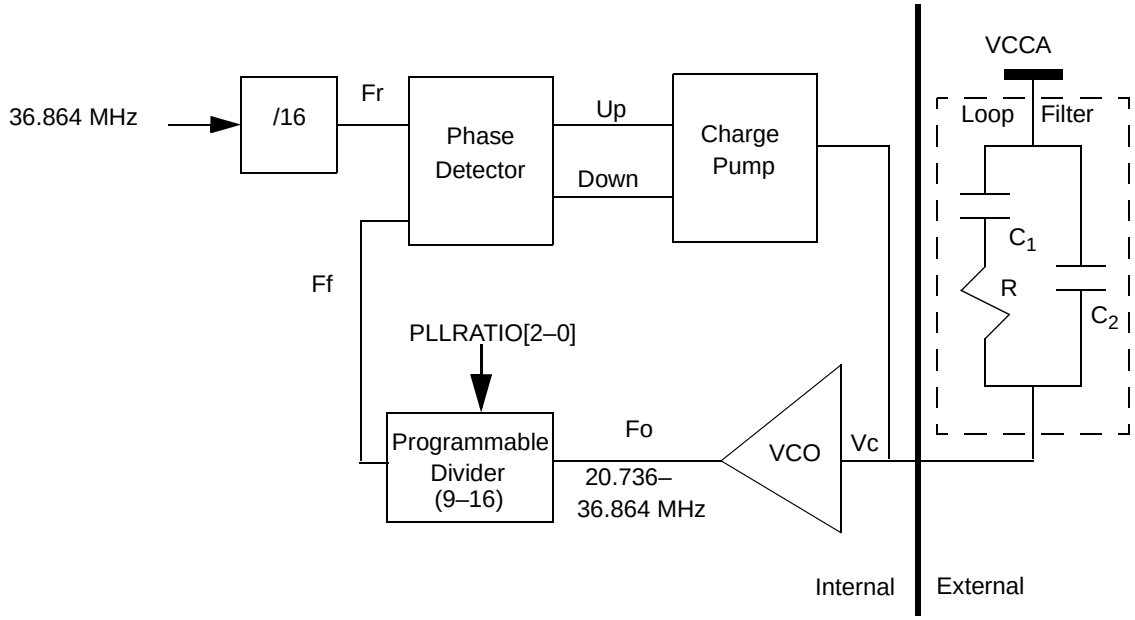


Figure 9. Graphics Dot Clock PLL Block Diagram

High-Speed PLL

The High-Speed PLL generates a 66.3552-MHz clock for the DRAM controller. Figure 10 on page 82 shows the block diagram for the High-Speed PLL. The input to the High-Speed PLL is the output of the Low-Speed PLL divided by five. The feedback divider is nine, which results in an output frequency (Fo) of 66.3552 MHz. This frequency is divided by 2 in the PMU to provide the 33-MHz input for the PLL in the CPU core.

Band Gap Block

The band gap reference circuit generates the bias currents for the four PLLs and provides a 2.4-V reference source for the RTC voltage monitor. The current sources, constant over V_{CC}, temperature, and process variations, are used by the four PLL charge pumps for adjusting the PLL operating frequency. The 2.4-V reference voltage is used by the RTC voltage monitor to detect a low backup battery voltage level.

RTC Voltage Monitor

The voltage monitor for the RTC block is shown in Figure 11 on page 82. Its functions are to provide a reset signal to the RTC block when it detects a low backup battery voltage, and to provide an early warning signal when the system is powering down.

The internal RTC reset signal is asserted on power-up if the back-up battery voltage drops below 2.4 V. The one shot prevents multiple resets during power-on.

An internal power-down signal is used by the RTC to isolate the RTC core from the rest of the microcontroller. The RTC voltage monitor uses the $\overline{\text{RESET}}$ assertion to detect a power-down. For proper operation, $\overline{\text{RESET}}$ and V_{CC} must follow the timing in Figure 12 on page 83.

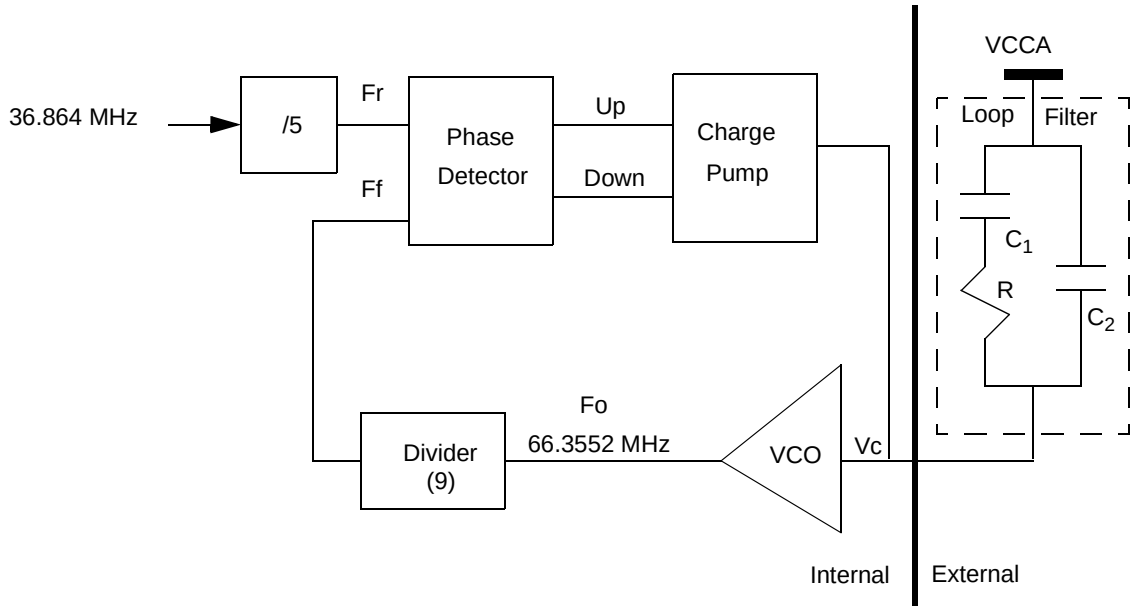


Figure 10. High-Speed PLL Block Diagram

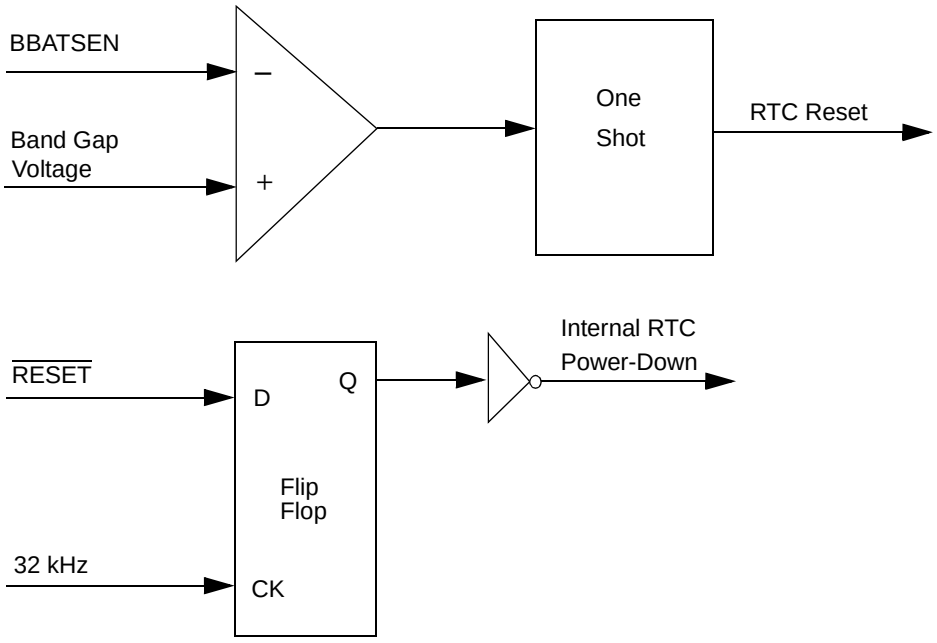
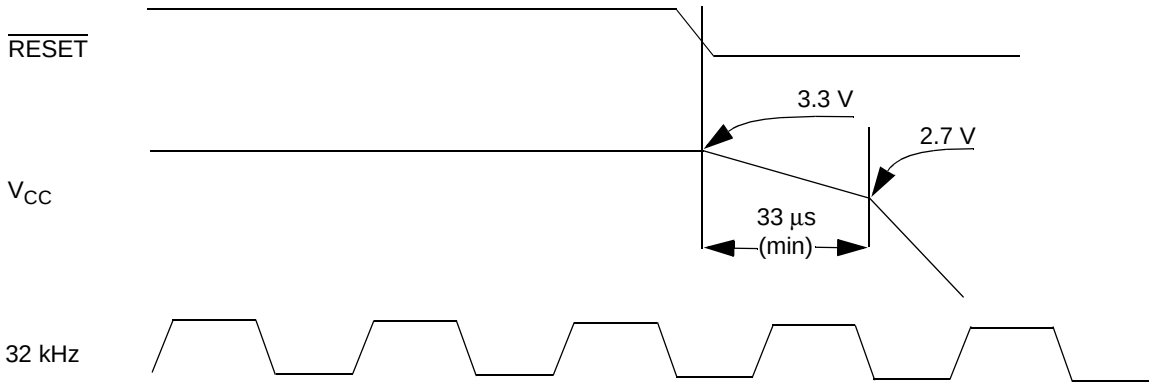


Figure 11. RTC Voltage Monitor Circuit



Notes:

1. *These timings apply only when powering down the chip while leaving only the RTC powered.*
2. *Applies to all V_{CC} except for the V_{CC}_RTC, which is left on for this mode.*
3. *Guarantees at least one rising edge after reset before 2.7 volts is reached.*

Figure 12. Timing Diagram for RTC-On Power-Down Sequence

Clock Specifications

The specifications for the external components required to implement the four PLL loop filters are shown in Table 28 on page 84.

Table 29 on page 84 lists the electrical specifications for the analog V_{CC} (VCCA) pin.

The on-chip crystal oscillator circuit supports most generic 32.768-kHz crystals as long as the specification for the crystals meet the electrical parameters listed in Table 30 on page 84.

The worst-case start-up time required for the PLLs is shown in Table 31 on page 84.

The PLL jitter specification is listed in Table 32 on page 85.

Programmable Interval Timer (PIT)

The ÉlanSC400 and ÉlanSC410 microcontrollers are equipped with a Programmable Interval Timer (PIT) that is software-compatible with PC/AT 8254 system timers. Historically, the clock source for this timer has been 1.19318 MHz. However, the internal PIT clock source is 1.1892 MHz. The user has two options:

- Use the internal PIT clock source (1.1892 MHz), which can adversely affect the Legacy software that depends on the 1.19318-MHz frequency.
- Drive an external 1.19318-MHz clock onto the CLK_IO pin and program this signal to be the source of the PIT clock.

For more details on this feature, refer to the subsection on configuring Timer Channel 0 in the programmable interval timer section of the *ÉlanSC400 and ÉlanSC410 Microcontrollers User's Manual*, order #21030.

Table 28. Loop-Filter Component Specification for PLLs

Parameter	Intermediate PLL	Low-Speed PLL	Graphics Dot Clock PLL	High-Speed PLL	Tolerance
C1	0.01 µF	470 pF	470 pF	330 pF	±10%
C2	0.001 µF	22 pF	33 pF	15 pF	±10%
R	4.7 KΩ	4.7 KΩ	4.7 KΩ	4.7 KΩ	±10%

Table 29. Analog V_{CC} (VCCA) Specification

Parameter	Min	Typ	Max	Unit
Peak-to-peak noise on VCCA			75	mV
Current consumption in High-Speed mode		2		mA
Current consumption in Low-Speed mode		2		mA
Current consumption in Standby mode		2		mA
Current consumption in Suspend mode (PLLs off)		1[1]		µA

Notes:
1. 2 mA if PLLs are enabled.

Table 30. 32.768-kHz Crystal Characteristics

Parameter	Min	Typ	Max	Unit
Nominal frequency		32.768		kHz
Load capacitance	13.5	15	16.5	pF
Q value	50			10^3
Series resistance			60	KΩ
Insulation resistance	100			MΩ
Shunt capacitance			2.5	pF

Table 31. Start-Up Time Specifications PLLs

Symbol	Parameter	Min	Typ	Max	Unit
t1	Intermediate PLL lock			10	ms
t2	Low-Speed PLL lock			100	µs
t3	High-Speed PLL lock			100	µs
t4	Graphics Dot Clock PLL lock			100	µs

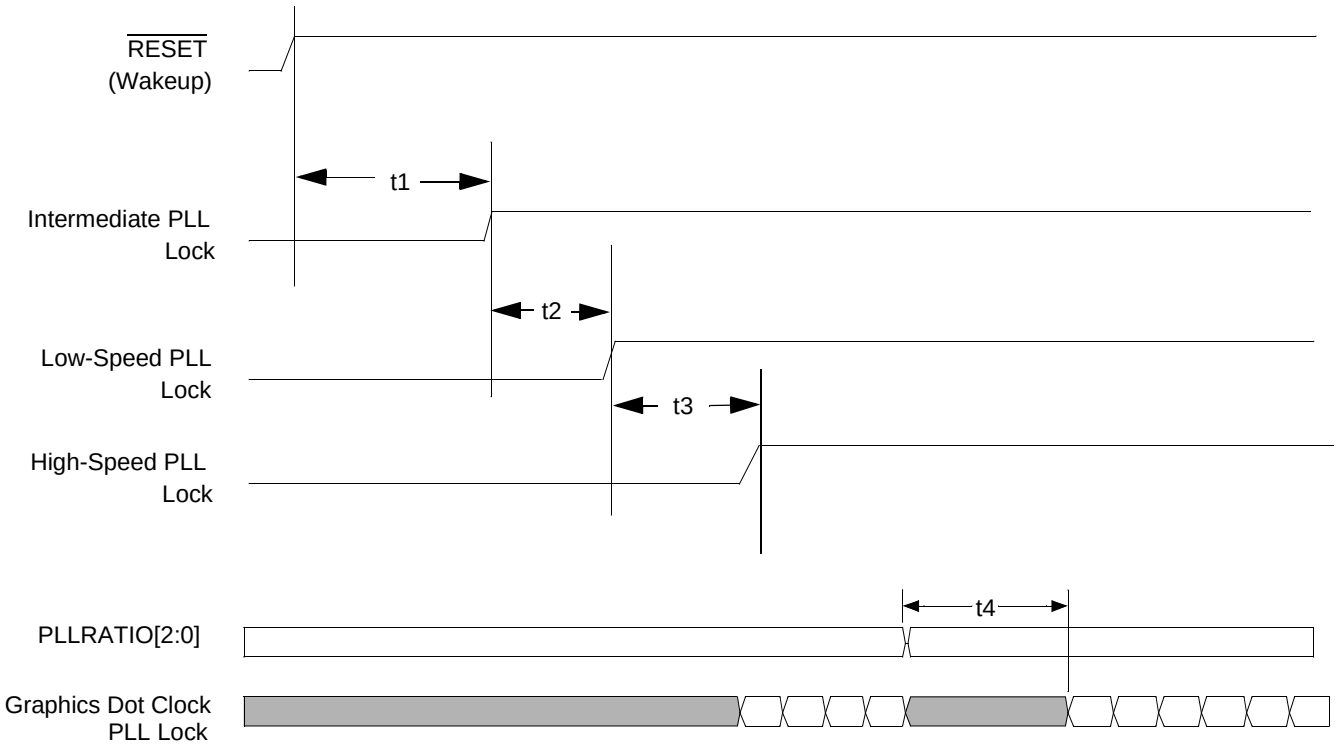


Figure 13. PLL Enabling Timing Sequence

Table 32. PLL Jitter Specification

PLL	Min	Typ	Max	Unit
Intermediate PLL frequency	1.4524	1.47456	1.4967	MHz
Intermediate PLL cycle-to-cycle jitter			20.4	ns
Low-Speed PLL frequency	36.311	36.864	37.417	MHz
Low-Speed PLL cycle-to-cycle jitter			0.82	ns
Graphics Dot Clock PLL frequency	–1.5%	Target	+1.5%	MHz
Graphics Dot Clock PLL cycle-to-cycle jitter			1	ns
High-Speed PLL frequency	65.360	66.3552	67.351	MHz
High-Speed PLL cycle-to-cycle jitter			0.5	ns

ABSOLUTE MAXIMUM RATINGS

Storage Temperature . . . . . .–65°C to +125°C

Ambient Temperature Under Bias . . –65°C to +110°C

Supply Voltage V_{CC} with Respect to GND . . . . . –0.5 V to +4.6 V

Voltage on 5-V-Tolerant Pins . . –0.5 V to (V_{CC} +2.6 V)

Voltage on Other Pins . . . . . .–0.5 V to (V_{CC} +0.5 V)

Stresses above those listed may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to Absolute Maximum Ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices

Ambient Temperature (T_A)
Operating in Free Air. . . . . .0°C to +70°C

Supply Voltage (V_{CC}) . . . . . +3.0 V to +3.6 V

CPU Voltage ($V_{CC_}$CPU) (33 & 66 MHz) +2.7 V to +3.6 V

CPU Voltage ($V_{CC_}$CPU) (100 MHz) +3.3 V to +3.6 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS OVER COMMERCIAL AND INDUSTRIAL OPERATING RANGES (BALL GRID ARRAY (BGA), 33 MHZ, 3.3 V)[1]

Symbol	Parameter Description	Min	Typ	Max	Unit
f_{OSC}	Frequency of Operation (internal CPU clock)	0		100	MHz
P_{CC}	Supply Power—CPU clock = 33 MHz ($V_{CC_}$CPU=3.3 V)		703	879	mW
P_{CCSS}[2]	Suspend Power at 3.3 V and 25°C—CPU idle, all internal clocks stopped except 32.768 kHz		264 (80 µA)	643.5 (195 µA)	µW
	Suspend Power at 3.3 V and 70°C—CPU idle, all internal clocks stopped except 32.768 kHz		726 (220 µA)	1950.5 (585 µA)	µW
P_{OFF}	RTC Power Only at 3.3 V		16	33	µW
$V_{OH(CMOS)}$	Output High Voltage $I_{OH(CMOS)}$ = –0.5 mA	V_{CC} – 0.45			V
$V_{OL(CMOS)}$	Output Low Voltage $I_{OL(CMOS)}$ = +0.5 mA			0.45	V
$V_{IH(CMOS)}$	Input High Voltage	2.0[3]		V_{CC} +0.3	V
$V_{IH(5\text{-}VTOL)}$	Input High Voltage (5-V tolerant inputs)	2.0[3]		V_{CC} +2.5	V
$V_{IL(CMOS)}$	Input Low Voltage	–0.3		+0.8	V
I_{LI}	Input Leakage Current (0.1 V ≤ V_{OUT} ≤ V_{CC}) (All pins except those with internal pullup/pulldown resistors)			±10	µA
I_{IH}	Input Leakage Current (V_{IH} = V_{CC} – 0.1 V) (All pins with internal pulldown resistors)			60	µA
I_{IL}	Input Leakage Current (V_{IL} = 0.1 V) (Pins with internal pullup resistors)			–60	µA
I_{LO}	Output Leakage Current (0.1 V ≤ V_{OUT} ≤ V_{CC}) (All pins except those with internal pullup/pulldown resistors)			±15	µA
$AVCC_{RP-P}$	Analog V_{CC} ripple peak to peak			100	mV

Notes:

1. V_{CCIO} = 3.0 V–3.6 V. For 33 and 66 MHz, T_{CASE} = 0°C to +95°C (commercial). T_{CASE} = –40°C to +95°C (industrial). For 100 MHz, T_{CASE} = 0°C to +85°C (commercial). Current out of a pin is given as a negative value.

2. In Suspend and Critical Suspend, the power state is the same. The PLLs are off, the LCD is disabled, and the CPU and all logic are in the lowest power state. The power management unit is active.

3. V_{CC} at 3.3 V.

Table 33. Operating Voltage (Commercial and Industrial)

Power Pin Type	33 MHz		66 MHz		100 MHz	
	Min	Max	Min	Max	Min	Max
Analog	2.7	3.6	2.7	3.6	2.7	3.6
CPU	2.7	3.6	2.7	3.6	3.3	3.6
RTC	2.7	3.6	2.7	3.6	2.7	3.6
V_{CC}	2.7	3.6	2.7	3.6	2.7	3.6

CAPACITANCE

Symbol	Parameter Descriptions	Test Conditions	Min	Max	Unit
C_{IN}	Input Capacitance	F_C = 1 MHz		15	pF
	Clock Capacitance			15	pF
C_{OUT}	Output Capacitance			20	pF
$C_{I/O}$	I/O Pin Capacitance			20	pF

Notes:

These parameters are not 100% tested, but are evaluated at initial characterization and at any time the design is modified where capacitance may be affected.

TYPICAL POWER NUMBERS

Power Requirements Under Different Power Management Modes

Table 34 shows the maximum and typical power dissipation for the ÉlanSC400 and ÉlanSC410 microcontrollers.

Table 34. Power Estimates

	Power Management Mode (CPU Clock Speed)					
	Hyper-Speed[1] (100 MHz)	**Hyper-Speed[1] (66 MHz)**	**High-Speed[2] (33 MHz)**	**Low-Speed[3] (~4 MHz)**	**Standby[4] (0 MHz)**	**Off[5]**
Maximum at 3.3 V	2194 mW (~665 mA)	1527 mW (~463 mA)	879 mW (~266 mA)	240 mW (~73 mA)	63 mW (~19 mA)	33 µW (10 µA)
Typical at 3.3 V	1818 mW (551 mA)	1222 mW (~370 mA)	703 mW (~213 mA)	192 mW (~58 mA)	50 mW (~15 mA)	16 µW (4.8 µA)
Maximum at 2.7 V	N/A	941 mW	586 mW	144 mW	60 mW	33 µW
Typical at 2.7 V	N/A	753 mW	469 mW	115 mW	48 mW	16 µW

Notes:

1. *Hyper-Speed mode is defined with a CPU clock frequency of 66 or 100 MHz. There is a time penalty to engage and disengage Hyper-Speed mode, because a CPU Stop Clock/Stop Grant sequence is required to "arbitrate" the internal CPU PLL start-up, cache flush, and the clearing of all internal pipelines and write buffers. The DX2 mode (66 MHz) is a clock-doubled mode with the CPU operating at 66 MHz and the rest of the system logic operating at 33 MHz. The DX4 mode (100 MHz) is a clock tripled mode with the CPU running at 100 MHz and the rest of the system running at 33 MHz.*
2. *High-Speed mode is defined with a maximum CPU clock speed of 33 MHz with a 1x dynamic clock-speed change control capability. Dynamic clock control allows fast, unarbitrated CPU clock-speed changes. Table 34 assumes a CPU frequency of 33 MHz and that the internal LCD controller is enabled. Other High-Speed mode power estimates with CPU V_{CC} = 3.3 V are shown below:*
   CPU Clock = 33 MHz/2 = 16.5 MHz, Max = 601 mW, Typical = 480 mW
   CPU Clock = 33 MHz/4 = 8.25 MHz, Max = 370 mW, Typical = 296 mW
3. *Low-Speed mode limits the maximum CPU clock frequency to 8 MHz. Table 34 assumes 8 MHz/2 = ~4.125 (CPU speed) and that the Internal LCD controller is enabled. Other Low-Speed power estimates with CPU at 3.3 V are shown below:*
   8 MHz/1 = 8.25 MHz, Max = 370 mW, Typical = 296 mW
   8 MHz/4 = 2.06 MHz, Max = 164 mW, Typical = 132 mW
4. *Standby mode is defined as having the CPU idle and stopped (0 MHz), but video screen refresh continues. IRQ0 (DOS Timer IRQ source) is assumed to be programmed as an activity and is generated at a rate of 60 Hz. This causes the PMU to transition to the Temporary Low-Speed mode where the CPU is clocked at 8 MHz. The assumed duration of the IRQ0 handler routine is 25 µs and, upon the interrupt return instruction, the PMU immediately re-enters the Standby mode, the LCD controller is enabled, and DRAM refresh type is slow CAS-before-RAS.*
5. *Off is defined as the V_{CC}_RTC supply pin having power applied and all other V_{CC} pins are not powered. In this mode, the core CPU, power management unit, PLLs, etc. have no power applied. The RTC will have an internally isolated power plane and source its power from the V_{CC}_RTC supply pin.*

DERATING CURVES

This section describes how to use the derating curves on the following pages to determine potential specified timing variations based on system capacitive loading. The Pin State Tables beginning on page 42 in this document have a column named “Max Load.” This column describes the specification load presented to the specific pin when testing was performed to generate the timing specification documented in the AC Characteristics section of this data sheet.

For example, to find out the effect of capacitive loading on a DRAM specification such as MWE hold from CAS Low, first find the specification load for MWE from Table 7 on page 44. The value here is 70 pF. Note the output drive type is programmable to C, D, or E. For this example, assume a drive strength of D, a system DRAM interface of 3.3 V, and a system load on the microcontroller’s MWE pin of 90 pF.

Referring to Figure 20, 3.3-V I/O Drive Type D Rise Time, on page 90, a time value of approximately 8.1 ns corresponds to a capacitive load of 70 pF.

Also referring to Figure 20, a time value of approximately 10 ns corresponds to a capacitive load of 90 pF. Subtracting 8.1 ns from the 10 ns, it can be seen that the rise time on the MWE signal will increase by 1.9 ns. Therefore, the MWE hold from CAS Low (min) parameter will increase from 30 ns to 31.9 ns (30 ns +1.9 ns).

If the capacitive load on MWE had been less than 70 pF, the time given in the derating curve for the load would be subtracted from the time given for the specification load. This difference can then be subtracted from the MWE hold from CAS Low (min) parameter to determine the derated AC timing parameter.

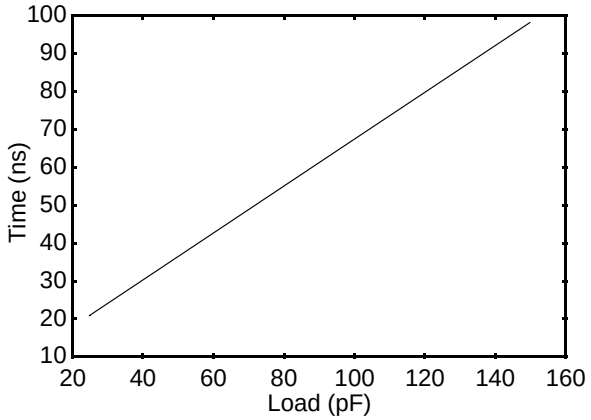


Figure 14. 3.3-V I/O Drive Type A Rise Time

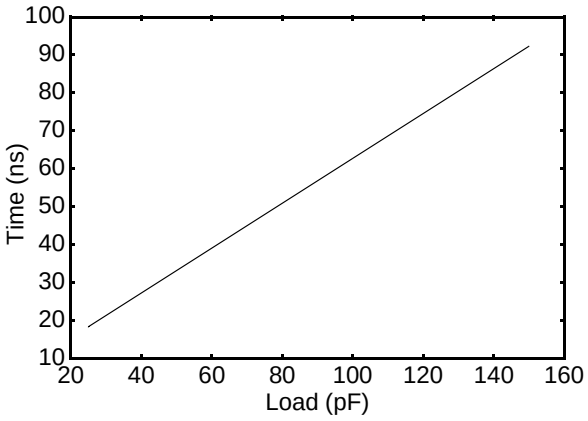


Figure 15. 3.3-V I/O Drive Type A Fall Time

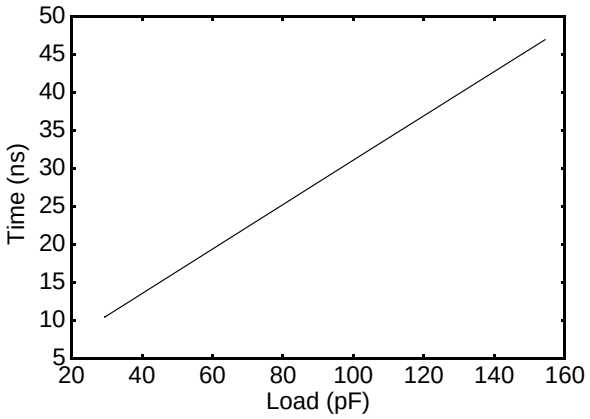


Figure 16. 3.3-V I/O Drive Type B Rise Time

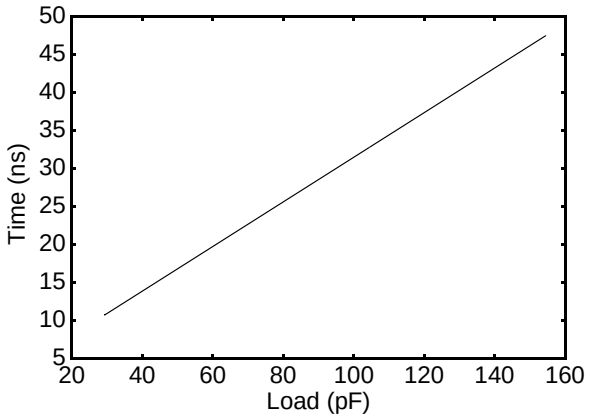


Figure 17. 3.3-V I/O Drive Type B Fall Time

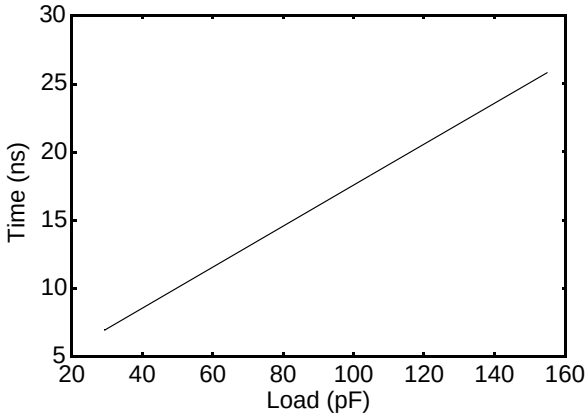


Figure 18. 3.3-V I/O Drive Type C Rise Time

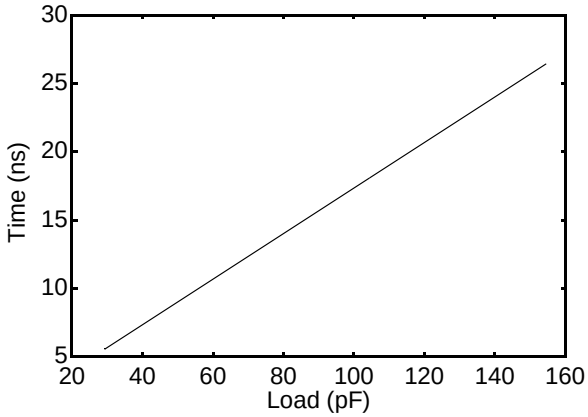


Figure 19. 3.3-V I/O Drive Type C Fall Time

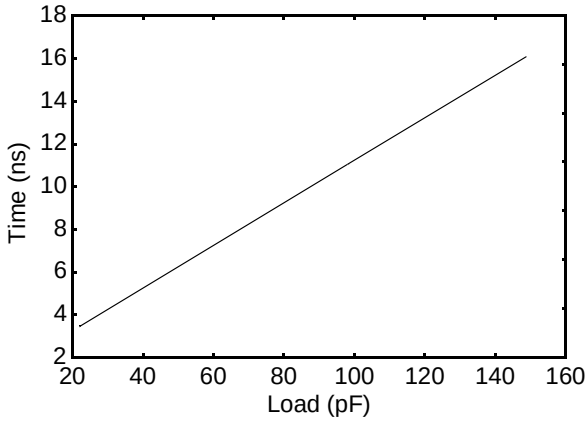


Figure 20. 3.3-V I/O Drive Type D Rise Time

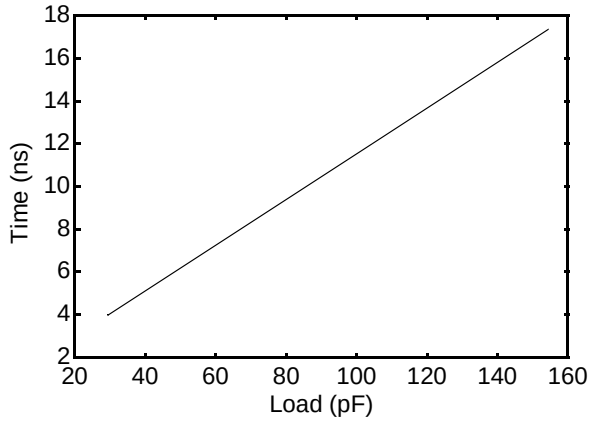


Figure 21. 3.3-V I/O Drive Type D Fall Time

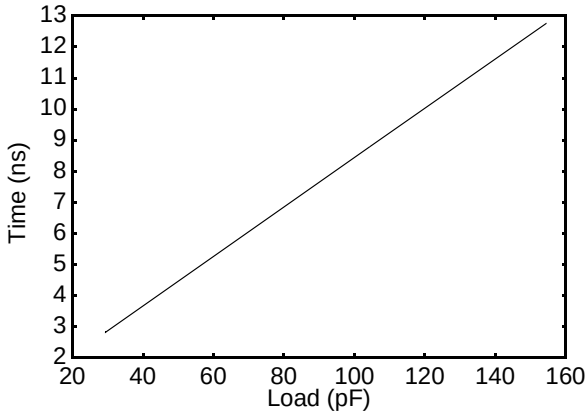


Figure 22. 3.3-V I/O Drive Type E Rise Time

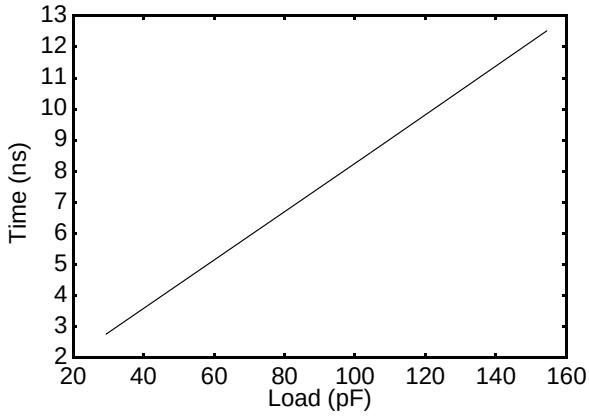


Figure 23. 3.3-V I/O Drive Type E Fall Time

AC SWITCHING CHARACTERISTICS AND WAVEFORMS

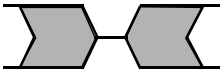
The AC specifications provided in the AC characteristics tables that follow consist of output delays, input setup requirements, and input hold requirements.

AC specifications measurement is defined by the figures that follow each timing table. All timings are referenced to 1.5 V unless otherwise specified.

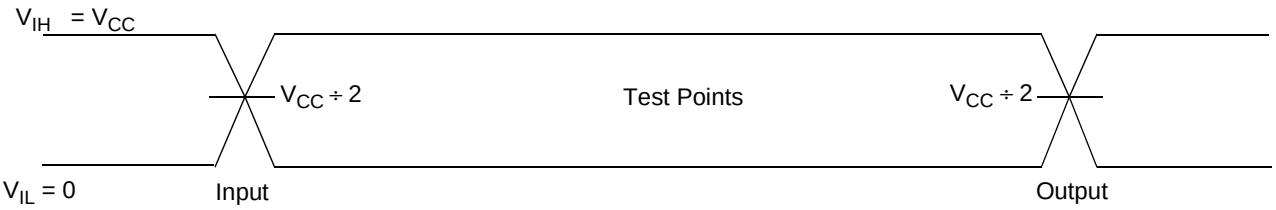
Output delays are specified with minimum and maximum limits, measured as shown. The minimum delay times are hold times provided to external circuitry.

Input setup and hold times are specified as minimums, defining the smallest acceptable sampling window. Within the sampling window, a synchronous input signal must be stable for correct microcontroller operation.

Key to Switching Waveforms

WAVEFORMS	INPUTS	OUTPUTS
	Must be Steady	Will be Steady
	May Change from H to L	Will be Changing from H to L
	May Change from L to H	Will be Changing from L to H
	Don’t Care, Any Change Permitted	Changing, State Unknown
	Does Not Apply	Center Line is High-Impedance "Off” State

AC SWITCHING TEST WAVEFORMS



Notes:

For AC testing, inputs are driven at 3 V for a logic 1 and 0 V for a logic 0.

Table 35. Power-On Reset Cycle

Symbol	Parameter Description	Notes	33-MHz External Bus			Unit
			Min	Typ	Max	
t1	V_{CC}_RTC valid hold before all other V_{CC}s are valid		0			s
t2	$\overline{\text{RESET}}$ valid hold from all V_{CC} valid (except V_{CC}_RTC)	1		0.5		s
t3	V_{CC}_RTC valid to BBATSEN active		100			µs
t4	CFGx setup to $\overline{\text{RESET}}$ inactive		5			µs
t5	CFGx hold from $\overline{\text{RESET}}$ inactive		0			ns
t6	RSTDRV pulse width		300			ms

Notes:

1. This parameter is dependent on the 32-kHz oscillator start-up time, which is dependent on the characteristics of the crystal, leakage and capacitive coupling on the board, and ambient temperature.

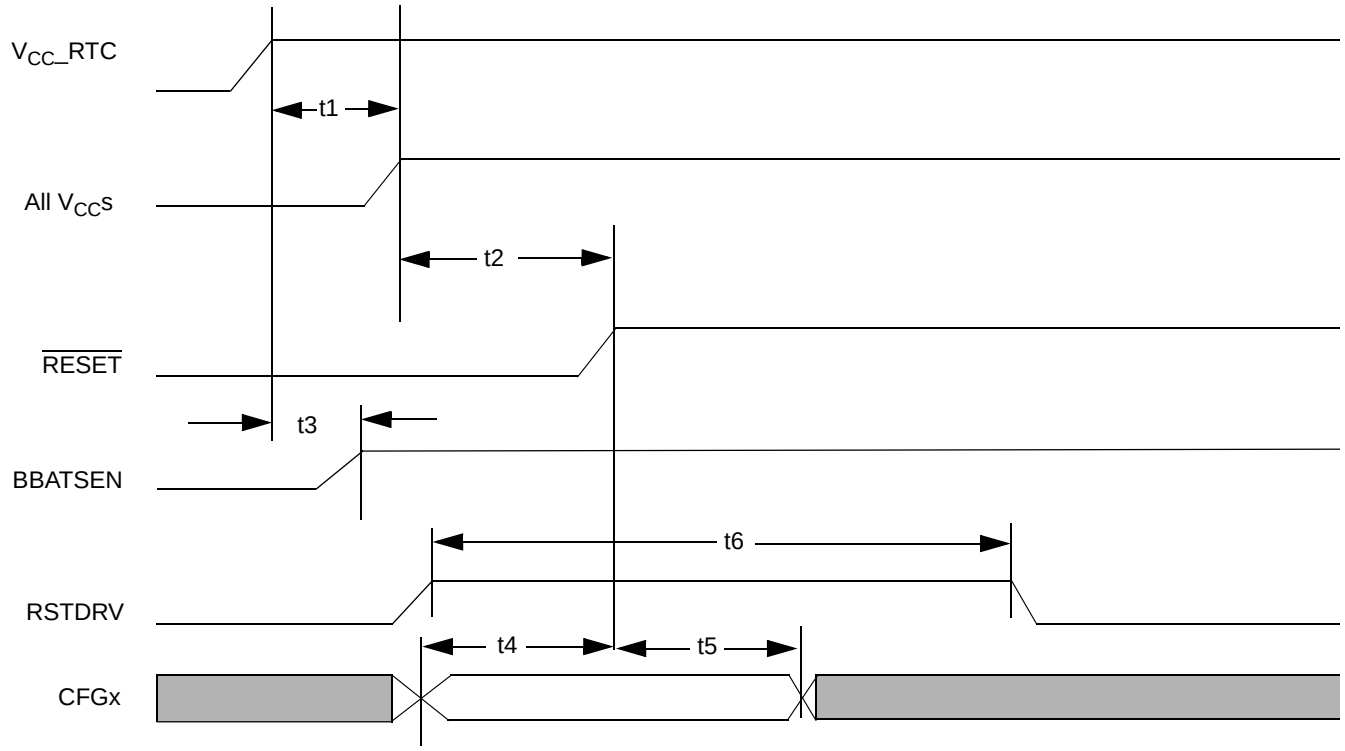


Figure 24. Power-Up Timing Sequence

Table 36. ROM/Flash Memory Cycles

Symbol	Parameter Description	Notes	33-MHz External Bus		Unit
			Min	Max	
t1	SA3–SA0 delay from SA31–SA4			6	ns
t2a	SA stable to ROMCSx assertion	1		6	ns
t2b	SA stable to ROMCSx assertion when qualified with command (ROMRD or ROMWR)	1	20		ns
t2c	SA stable to ROMCSx assertion when qualified with command (ROMRD or ROMWR)	1	100		ns
t3	ROMCSx deassertion to SA change	1	53		ns
t4	SD setup to ROMRD or ROMCSx deassertion or burst address switching, whichever is first, for 8-/16-/32-bit device	2	15		ns
t5	ROMWR setup to ROMCSx		0		ns
t6	Data hold from SA, ROMRD, or ROMCSx change, whichever is first	2	0		ns
t7	ROMCSx pulse width	3	25		ns
t8	DBUFOE, R32BFOE setup to ROMRD, ROMWR Low			-8	ns
t9	ROMRD pulse width	3	25		ns
t10	SA3–SA0 burst address valid duration	3	25		ns
t11	ROMWR pulse width	3	25		ns
t12	SD setup to ROMWR assertion for 32-bit device	2		17	ns
t13	SD hold from ROMWR deassertion	2	20		ns
t14	SA hold from ROMWR deassertion	2	20		ns
t15	ROMRD delay from SA stable		115		ns
t16a	ROMRD, ROMWR pulse width for 8-bit device		530		ns
t16b	ROMRD, ROMWR pulse width for 16-bit device		240		ns
t17a	Data setup from ROMRD for 8-bit device	2, 4		489	ns
t17b	Data setup from ROMRD for 16-bit device	2, 4		209	ns
t18	ROMRD deassertion to SA unstable		20		ns
t19	Data hold from ROMRD deassertion	2	0		ns
t20	SA hold from ROMWR deassertion		53		ns
t21a	SD setup to ROMWR assertion for 16-bit device	2	-29		ns
t21b	SD setup to ROMWR assertion for 8-bit device	2	33		ns
t22	SD hold from ROMWR deassertion	2	26		ns
t23	IOCHRDY assertion to ROMRD, ROMWR deassertion		125		ns
t24a	IOCHRDY deassertion from ROMRD, ROMWR for 8-bit			378	ns
t24b	IOCHRDY deassertion from ROMRD, ROMWR for 16-bit			66	ns
t25	R32BFOE/DBUFOE hold from ROMRD High		0		ns
t26	R32BFOE/DBUFOE hold from ROMWR High		26		ns
t27	DBUFRDL, DBUFRDH setup to ROMRD, ROMWR Low			-8	ns
t28	DBUFRDL, DBUFRDH hold from ROMRD High		0		ns
t29	DBUFRDL, DBUFRDH hold from ROMWR high		26		ns

Notes:

1. *The ROMCSx address decode is programmable for an early decode (via bit 5 in CSC index 23h, 25h, and 27h). The early address-decode is available to provide the ROMCSx by qualifying the address signals only; it is not qualified with the commands (ROMRD, ROMWR). The timing parameter t2a pertains to the early address-decode feature being enabled (ROMCSx is address-decode only). Parameters t2b and t2c are observed when the early address-decode feature is disabled (ROMCSx is address-decode qualified with command). The early decode can be enabled for both Fast-mode and Normal-mode ROM accesses.*
2. *When a x32 DRAM or VL bus is enabled, additional delay must be added to accommodate for the delay through the external data buffers required for the SD bus in this mode.*
3. *There are two types of programmable wait states. The first programmable wait state is always used in the first access for either burst or non-burst supported device. It starts at the assertion of the chip select or at the transition of SA3–SA0, whichever occurs later. The second programmable wait state is used only for any subsequent burst read accesses to a burst mode ROM device. It starts at the transition of SA3–SA0. The burst address valid duration depends on which wait state is used. If the wait state is set to zero, then the minimum address duration is 30 ns (one bus clock cycle).*
4. *If wait states are added via the deassertion of IOCHRDY, the data setup time to IOCHRDY assertion is 0 ns (minimum).*

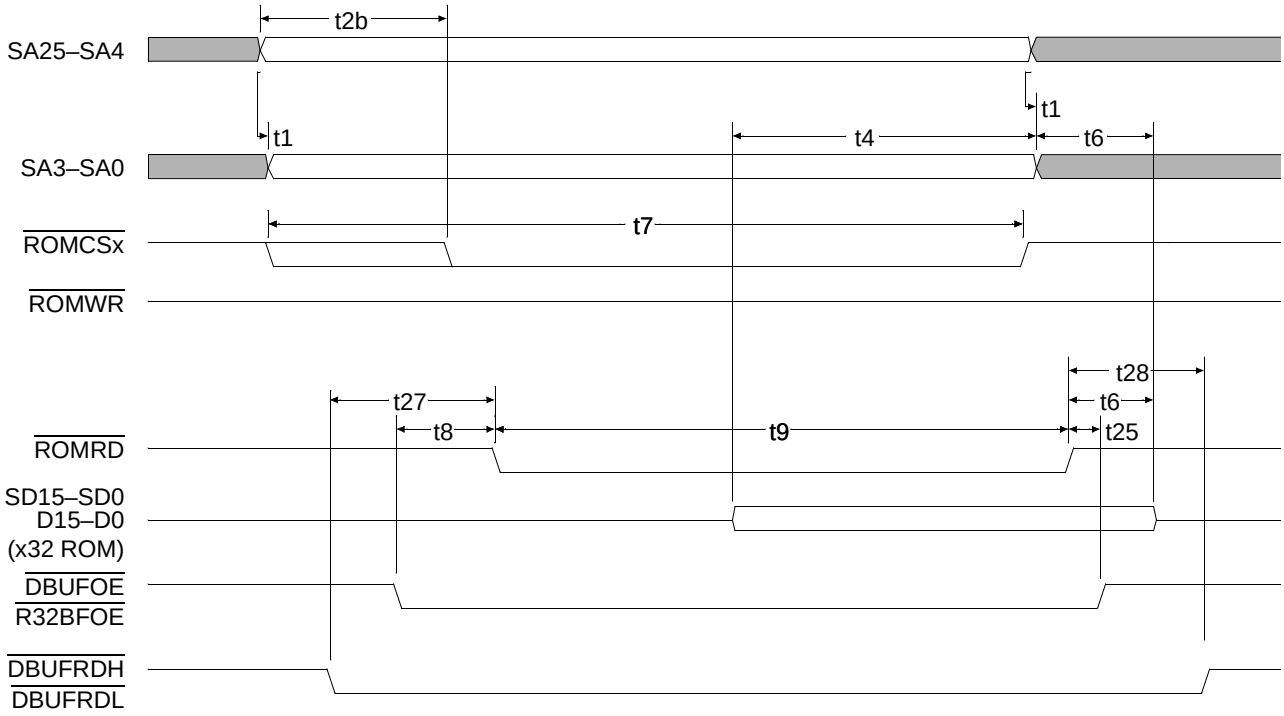
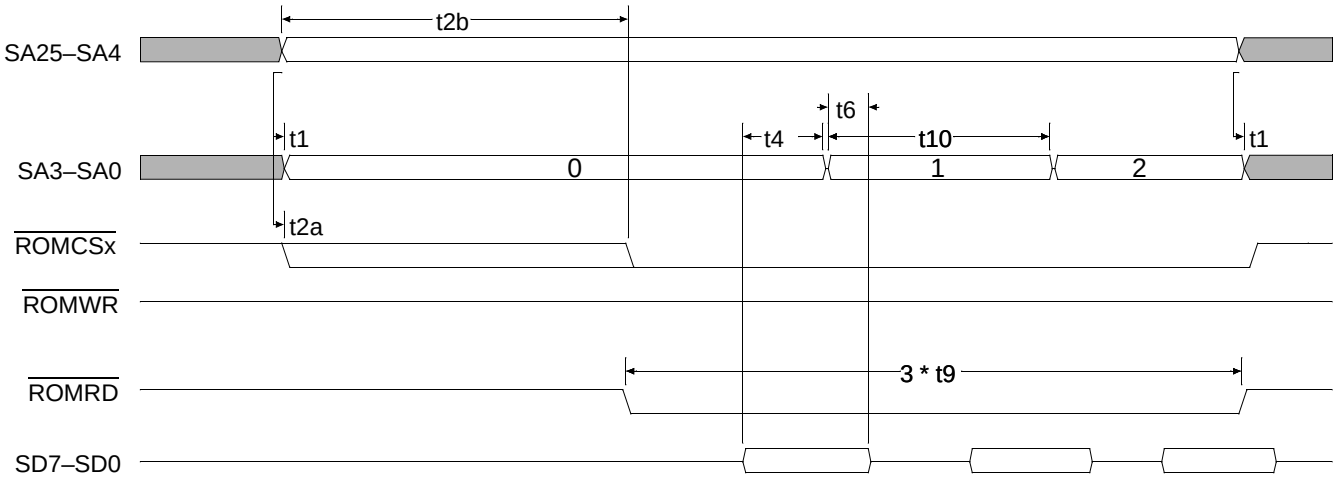


Figure 25. Fast Mode 8-/16-/32-Bit ROM/Flash Memory Read Cycle



Notes:

The ROM controller fetches the number of bytes requested by the CPU as dictated by the CPU $\overline{BE}$ (Byte Enable) signals and returns the data as a single transfer. In this example, $\overline{BE}$ was set to "0001". Therefore, the ROM controller generates additional addresses to read all three bytes before returning them to the CPU.

Figure 26. Fast Mode CPU Read of Three Consecutive Bytes from 8-Bit ROM/Flash Memory

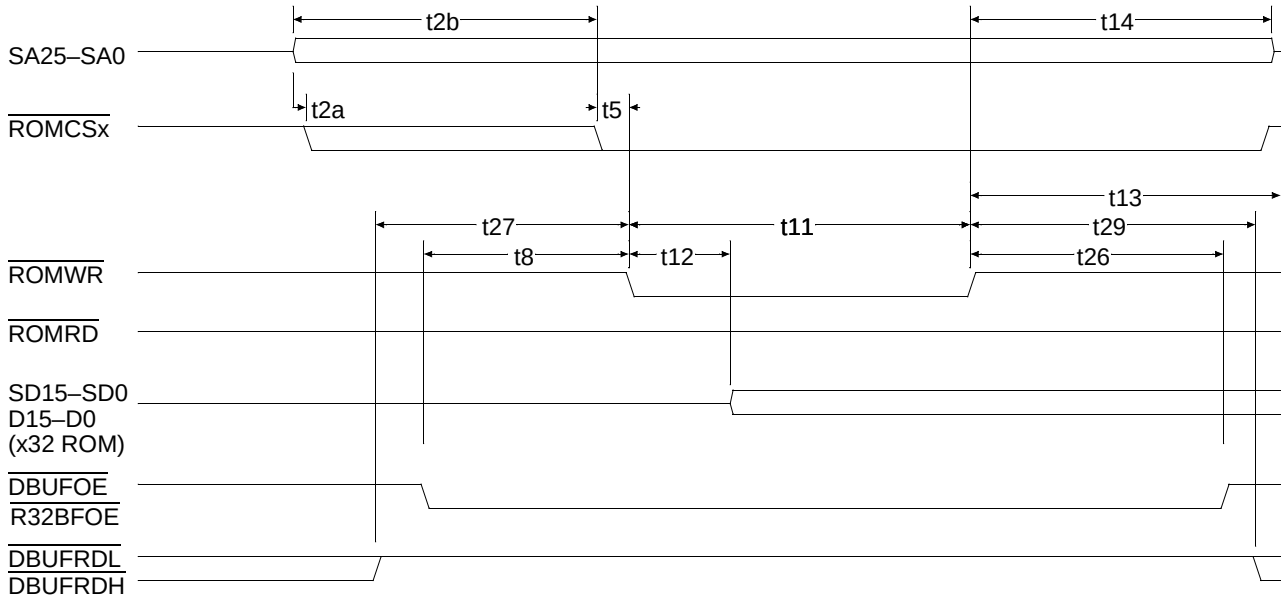


Figure 27. Fast Mode 8-/16-/32-Bit Flash Memory Write Cycles

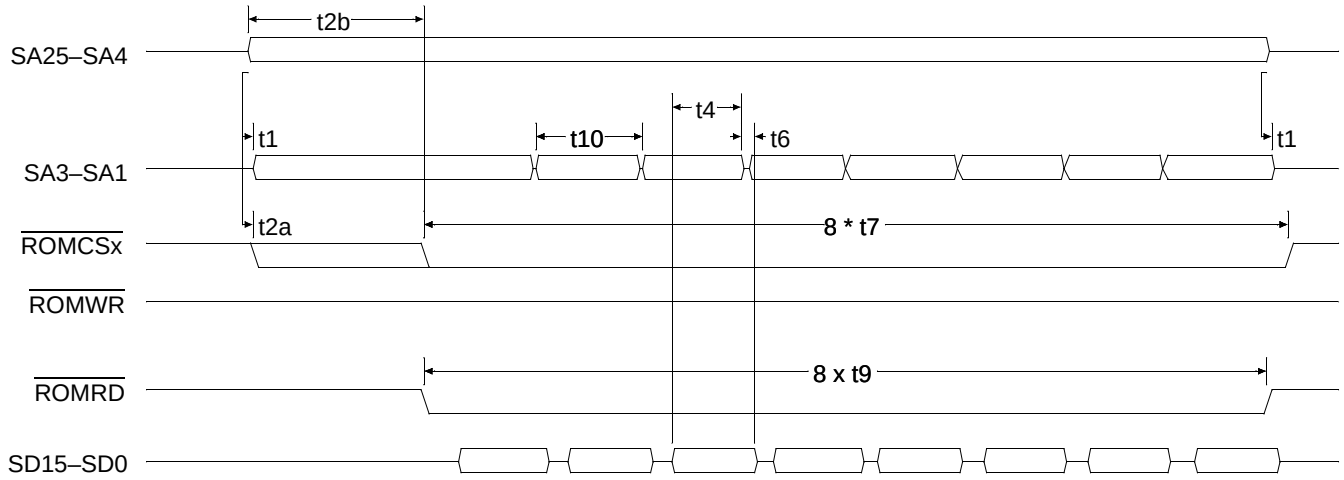


Figure 28. Fast Mode 16-Bit Burst ROM Read Cycles

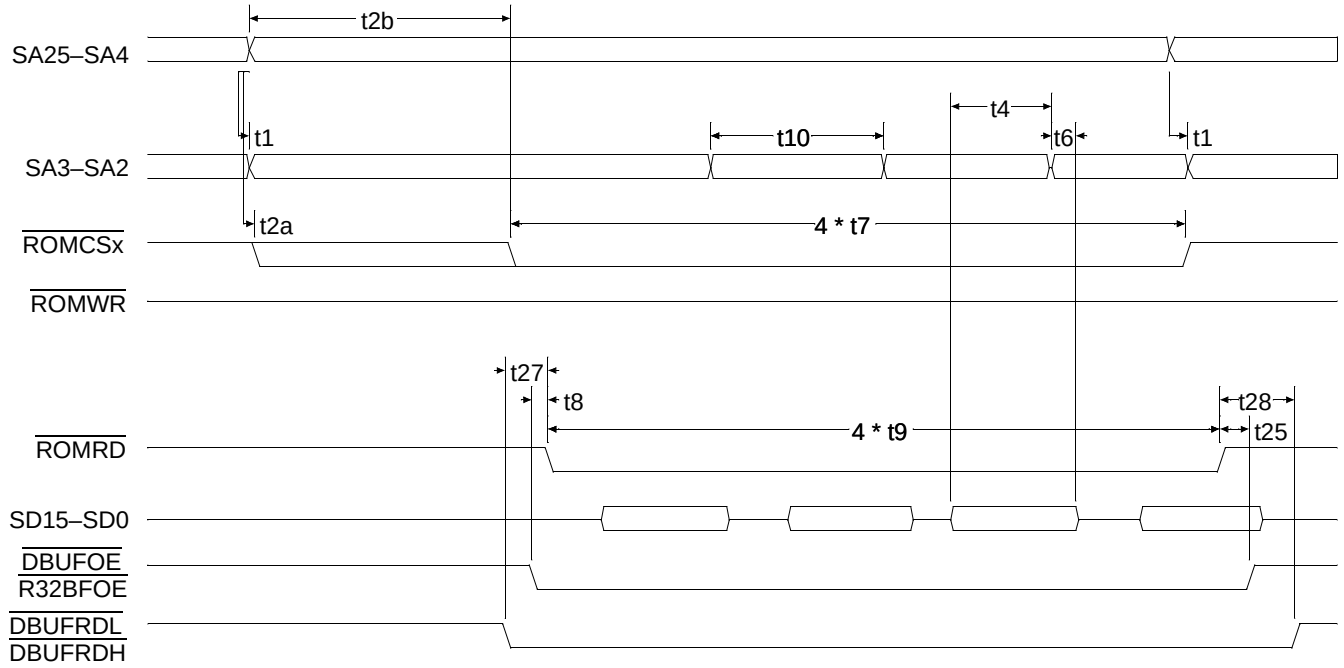


Figure 29. Fast Mode CPU Burst Read from 32-Bit Burst Mode ROM/Flash Memory

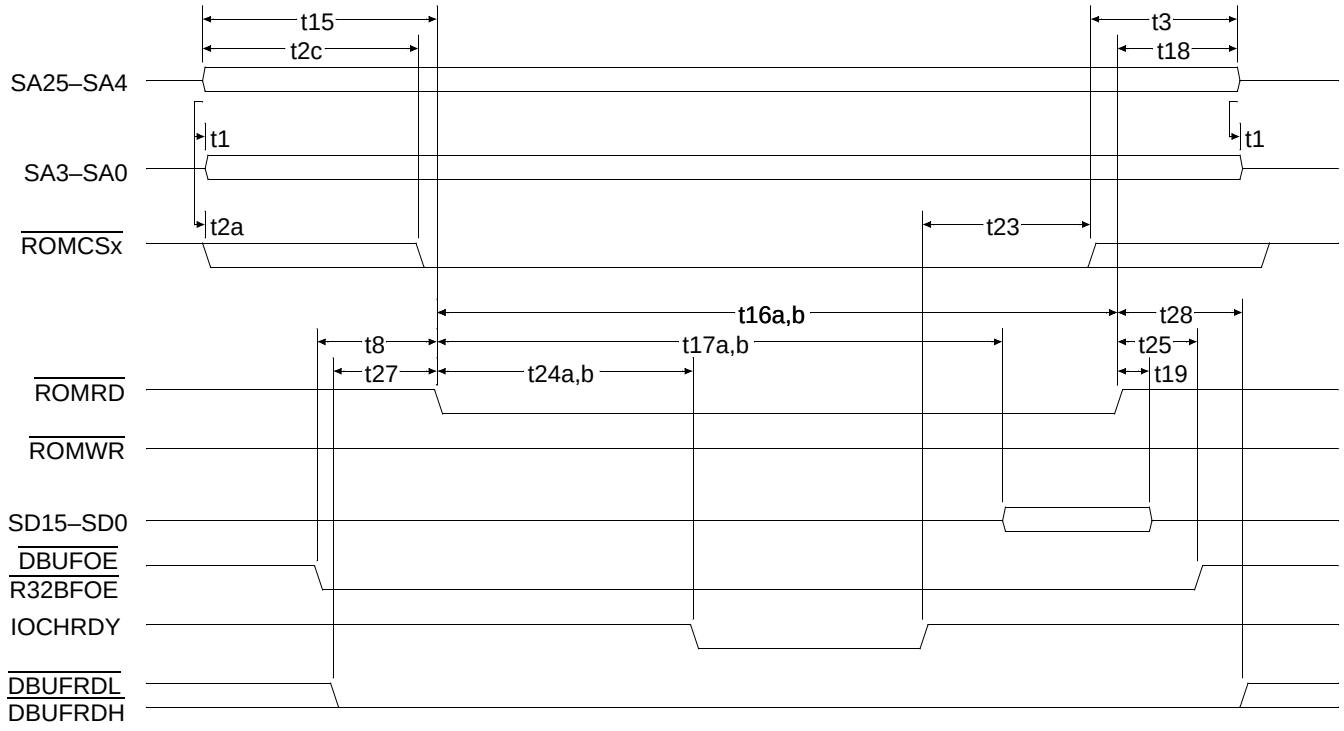


Figure 30. Normal Mode 8-/16-Bit ROM/Flash Memory Read Cycles

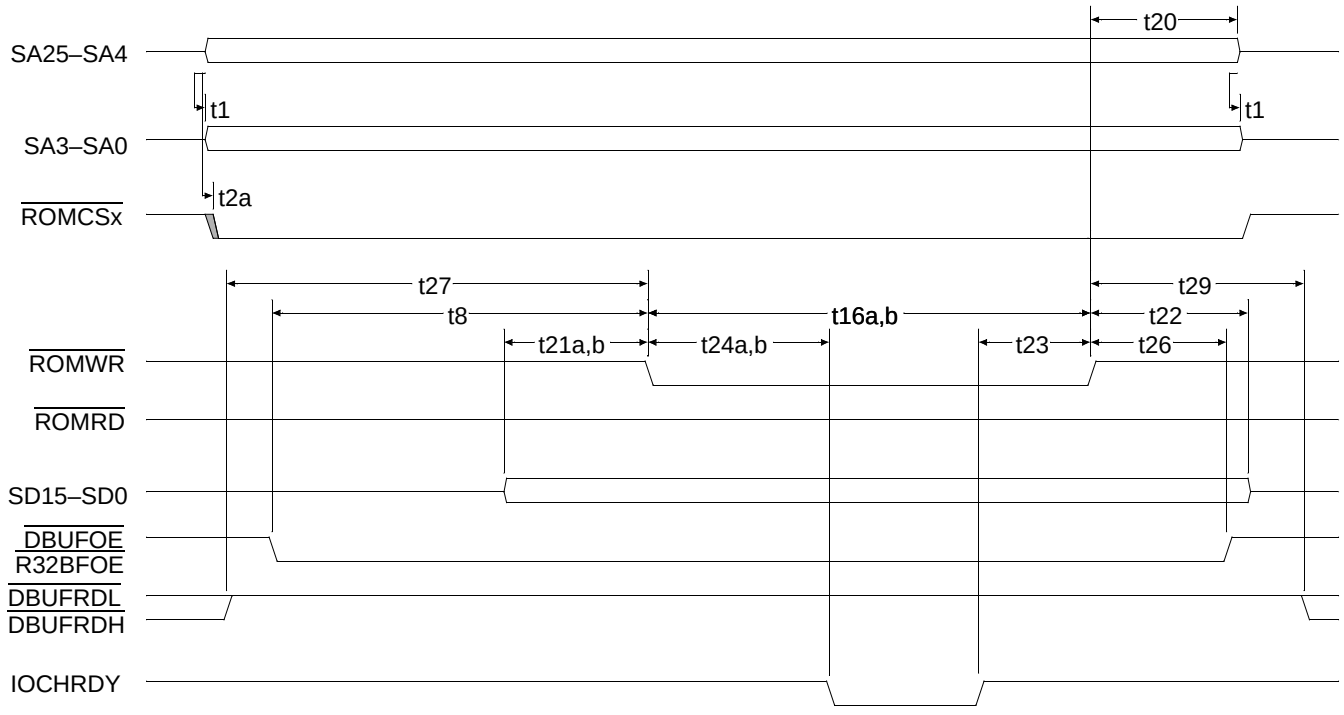


Figure 31. Normal Mode 8-/16-Bit Flash Memory Write Cycles

Table 37. DRAM Cycles

Symbol	Parameter Description	Notes	33-MHz External Bus		Unit
			Min	Max	
t1	Row address setup time	1	5		ns
t2	$\overline{RAS}$ to $\overline{CAS}$ delay	1	42.75		ns
t3	Row address hold time	1	14.25		ns
t4	Column address setup time	1	0		ns
t5	Column address hold time	1	14.25		ns
t6a	$\overline{CAS}$ pulse width (CPU, Fast Page mode)	1	42.75		ns
t6a	$\overline{CAS}$ pulse width (graphics controller, Fast Page mode)	1	28.5		ns
t6b	$\overline{CAS}$ pulse width (EDO mode)	1	28.5		ns
t7a	$\overline{CAS}$ precharge (non-interleaved)	1	14.25		ns
t7b	$\overline{CAS}$ precharge (interleaved)	1	71.25		ns
t8a	$\overline{CAS}$ hold	1	85.5		ns
t8b	$\overline{CAS}$ hold (EDO)	1	66.5		ns
t9a	Fast page mode cycle time (non-interleaved)	1	57		ns
t9b	Fast page mode cycle time (interleaved)	1	114		ns
t9c	EDO mode cycle time	1	57		ns
t10	Access time from $\overline{RAS}$	1	66.5		ns
t11	Access time from column address	1	35		ns
t12a	Access time from $\overline{CAS}$	1	20		ns
t12b	Access time from $\overline{CAS}$ (EDO)	1	22		ns
t13	Access time from $\overline{CAS}$ precharge	1	40		ns
t14a	Read data hold from $\overline{CAS}$	1	0		ns
t14b	Read data hold from $\overline{CAS}$ (EDO)	1	5		ns
t15	MA12–MA0 switching time	1		15	ns
t16	Delay between bank $\overline{CAS}$ signals	1		15	ns
t17	$\overline{MWE}$ setup to $\overline{CAS}$	1	10		ns
t18	$\overline{MWE}$ hold from $\overline{CAS}$	1	30		ns
t19	Write data setup to $\overline{CAS}$	1	10		ns
t20	Write data hold from $\overline{CAS}$	1	30		ns
t21	$\overline{RAS}$ precharge	1	60		ns
t22	$\overline{RAS}$ pulse width	1	75		ns
t23	$\overline{RAS}$ hold	1	28.5		ns
t24	$\overline{MWE}$ low from $\overline{CAS}$ (EDO data disable)	1	14.25		ns
t25	$\overline{MWE}$ pulse width (EDO)	1	14.25		ns
t26	Data high impedance from $\overline{MWE}$	1		15	ns
t27	$\overline{RAS}$ to $\overline{CAS}$ precharge time	1	15		ns
t28	$\overline{CAS}$ setup time ($\overline{CAS}$-before-$\overline{RAS}$ refresh)	1	10		ns
t29	$\overline{CAS}$ hold time ($\overline{CAS}$-before-$\overline{RAS}$ refresh)	1	25		ns
t30	$\overline{RAS}$ pulse width during self-refresh cycle	1	100		us
t31	$\overline{RAS}$ precharge time during self-refresh cycle	1	130		ns
t32	$\overline{WE}$ setup time ($\overline{CAS}$-before-$\overline{RAS}$ refresh)	1	25		ns
t33	$\overline{WE}$ hold time ($\overline{CAS}$-before-$\overline{RAS}$ refresh)	1	25		ns

Notes:

1. All timings assume 70-ns DRAMs, fastest programmable timing, and a 66-MHz clock for the memory controller.

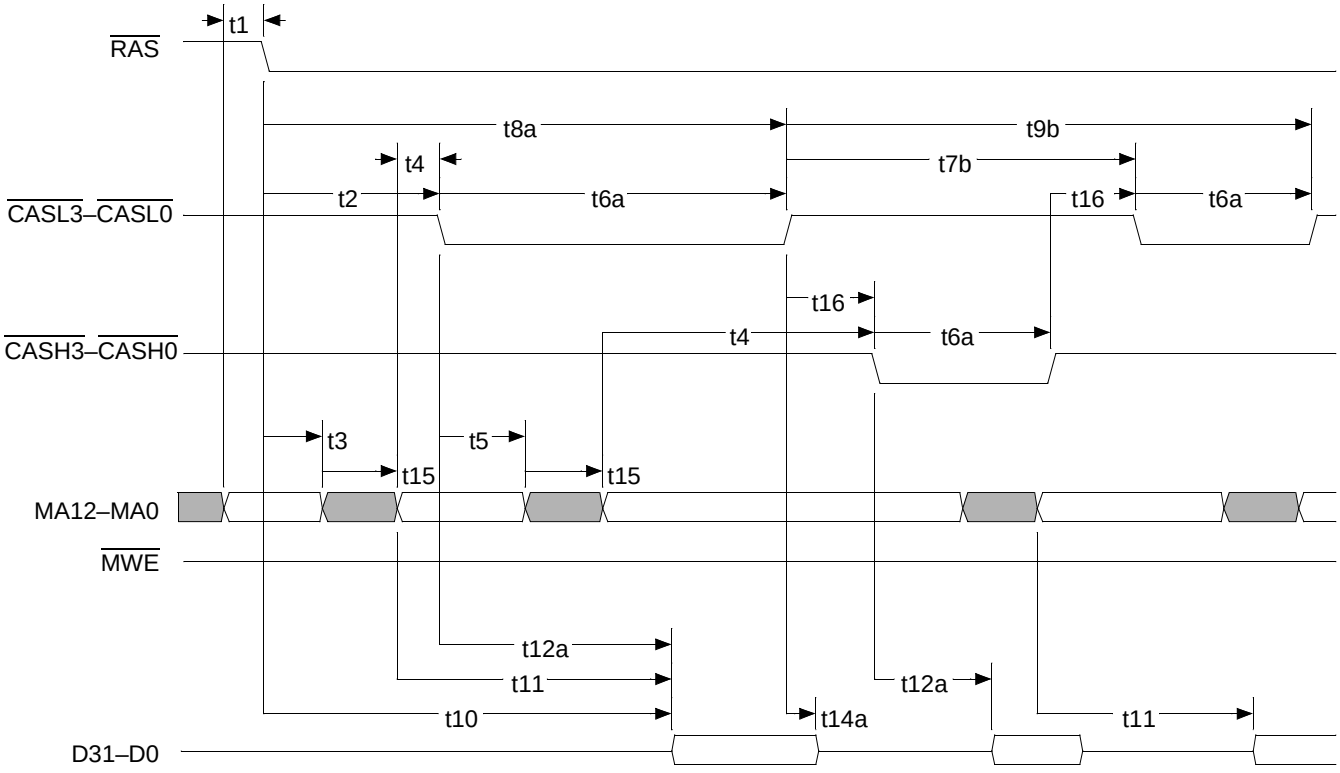


Figure 32. DRAM Page Hit Read, Interleaved

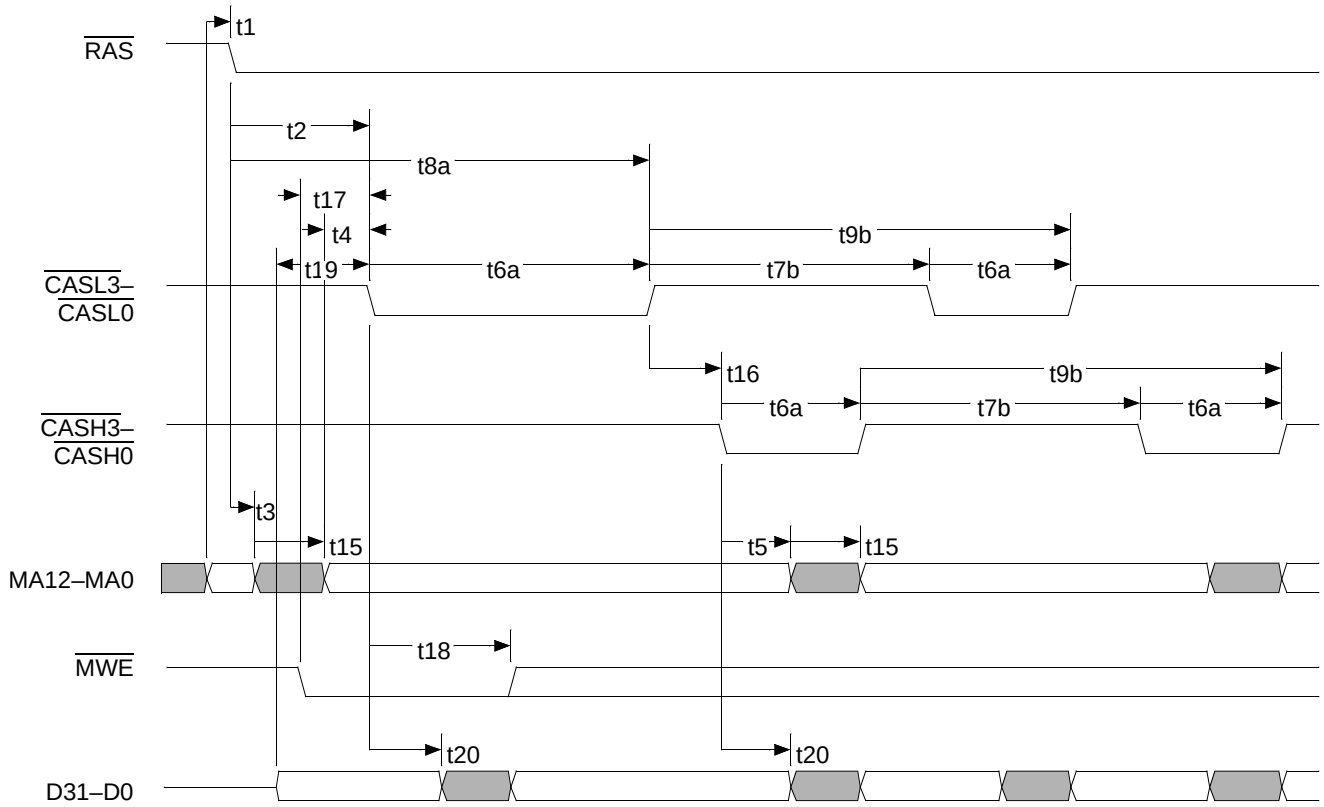


Figure 33. DRAM Page Hit Write, Interleaved

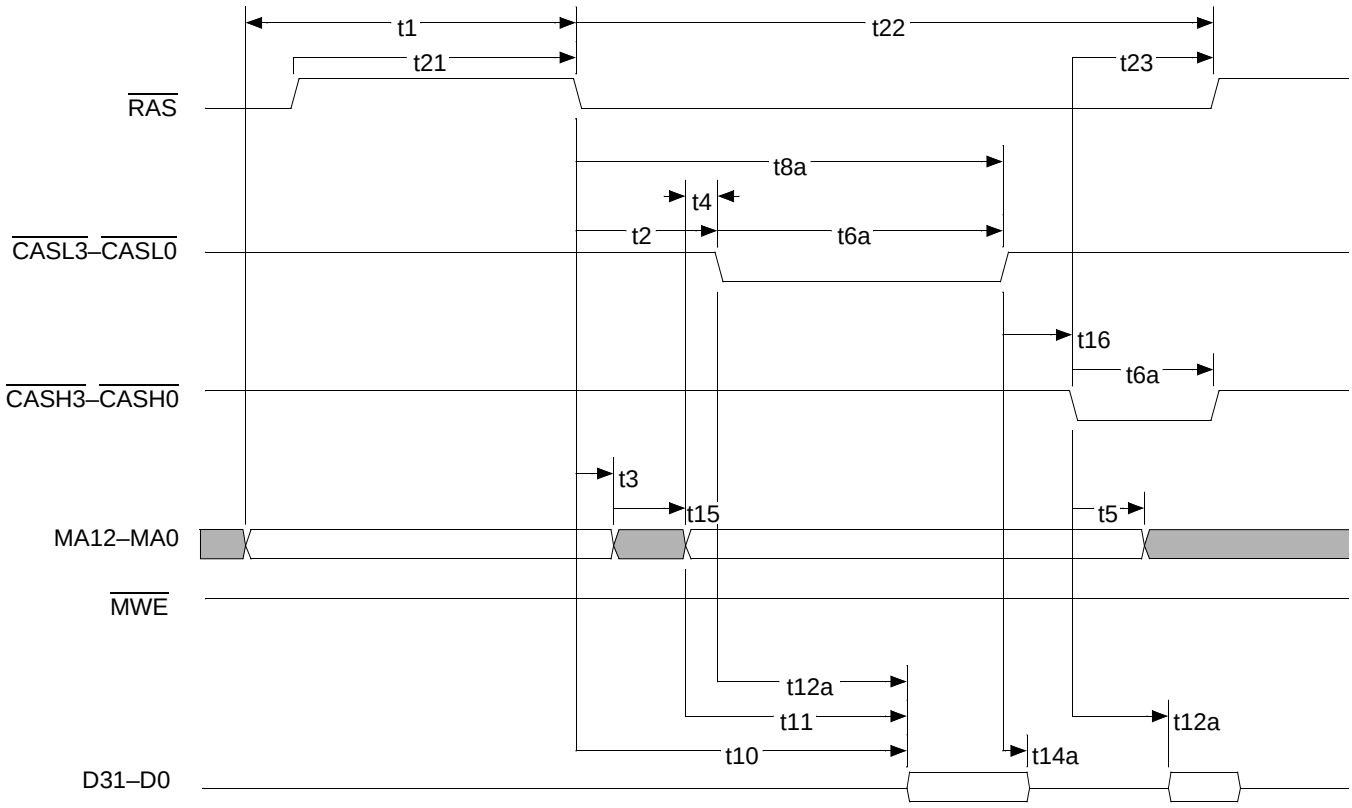


Figure 34. DRAM Page Miss Read, Interleaved

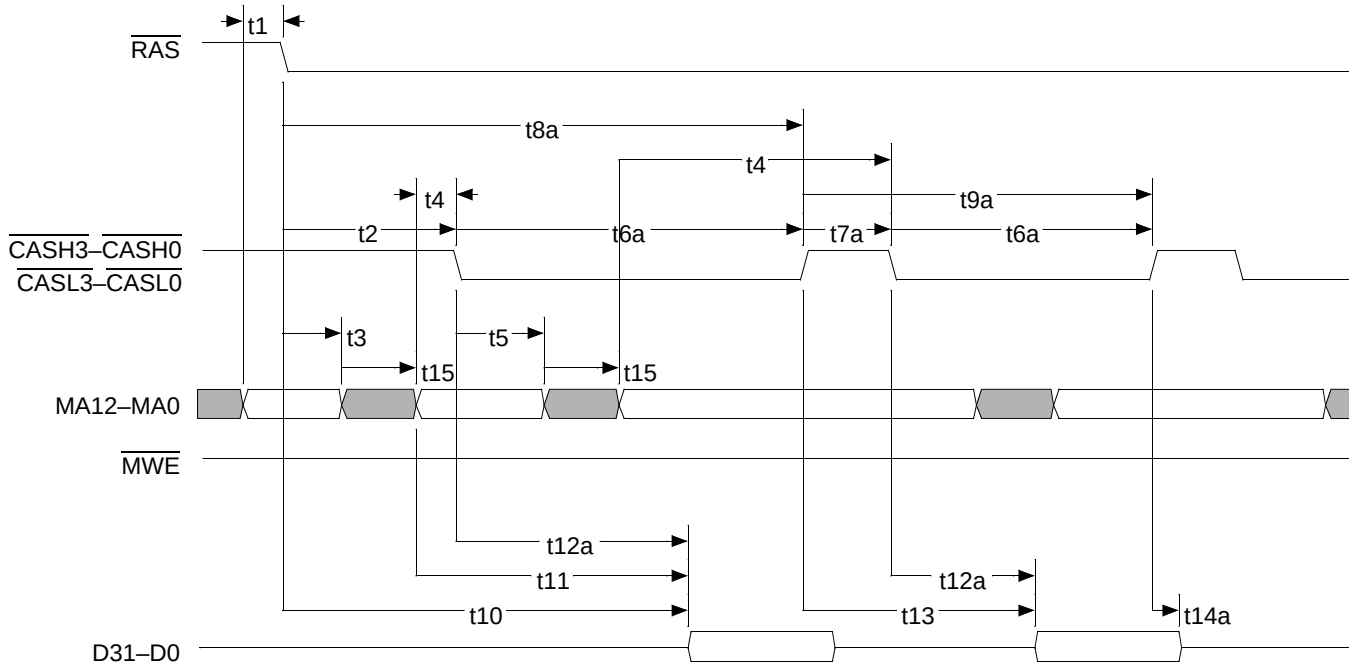


Figure 35. DRAM Page Hit Read, Non-Interleaved

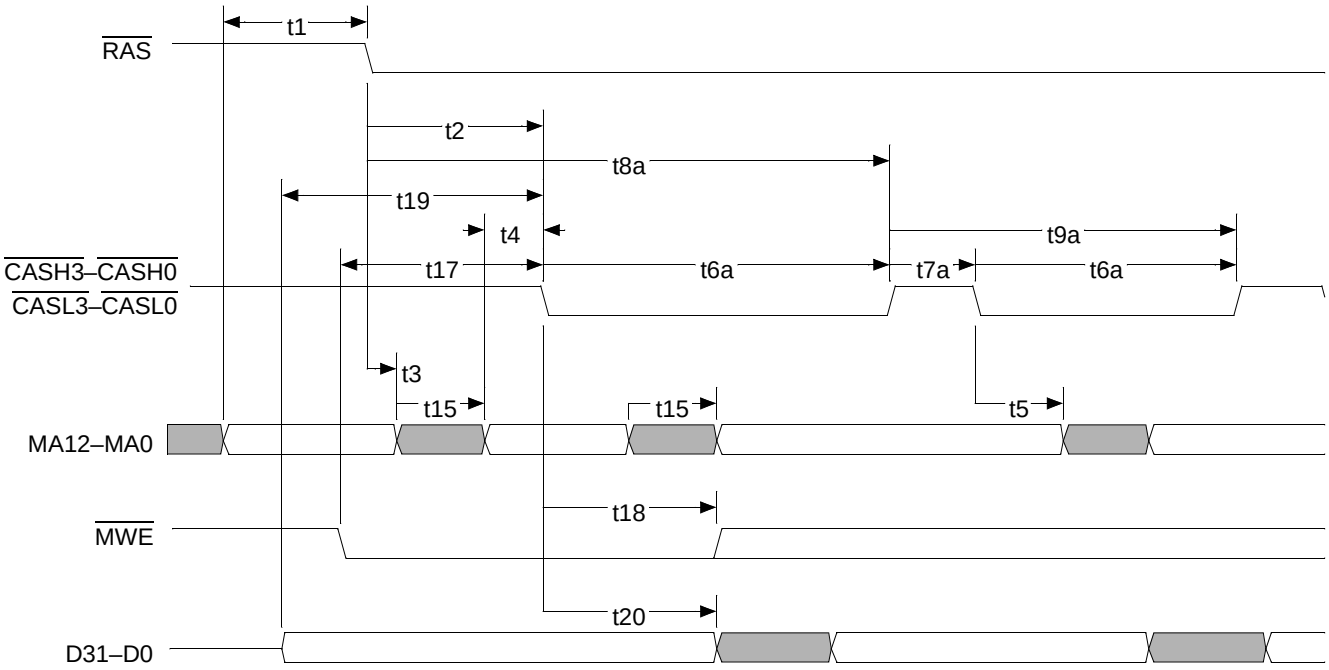


Figure 36. DRAM Page Hit Write, Non-Interleaved

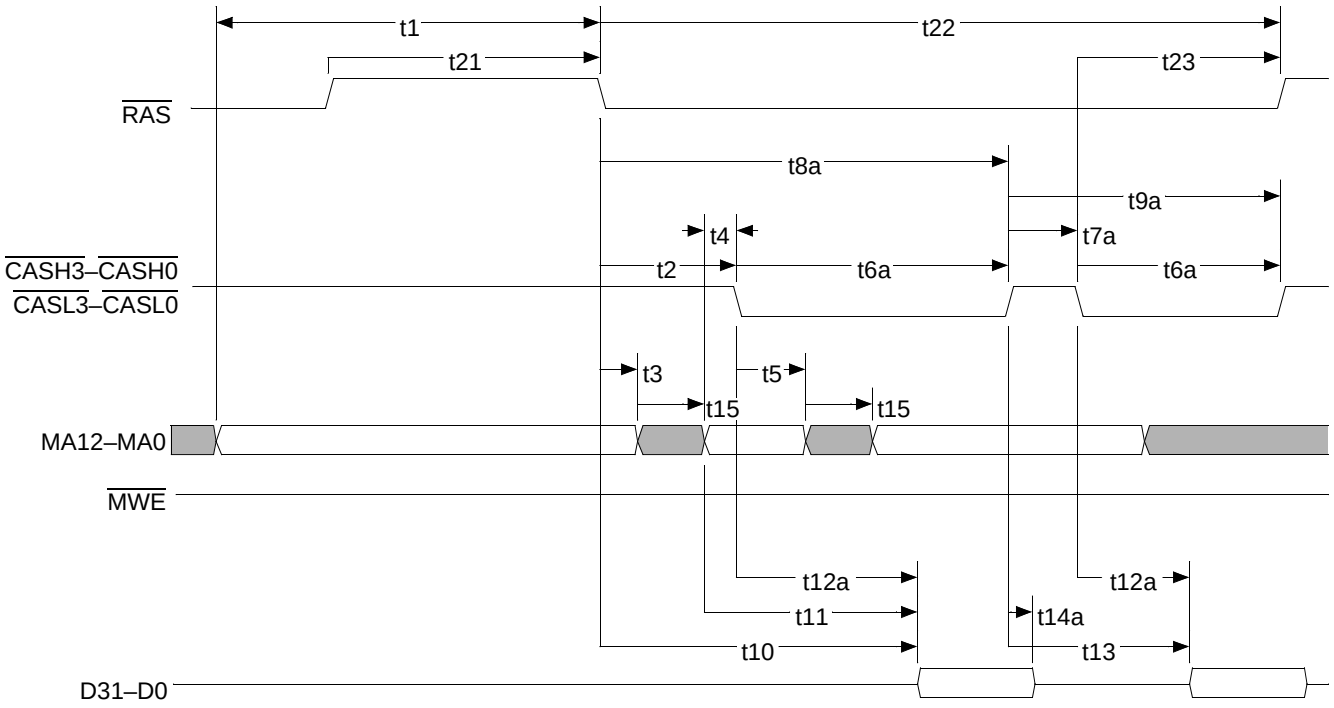
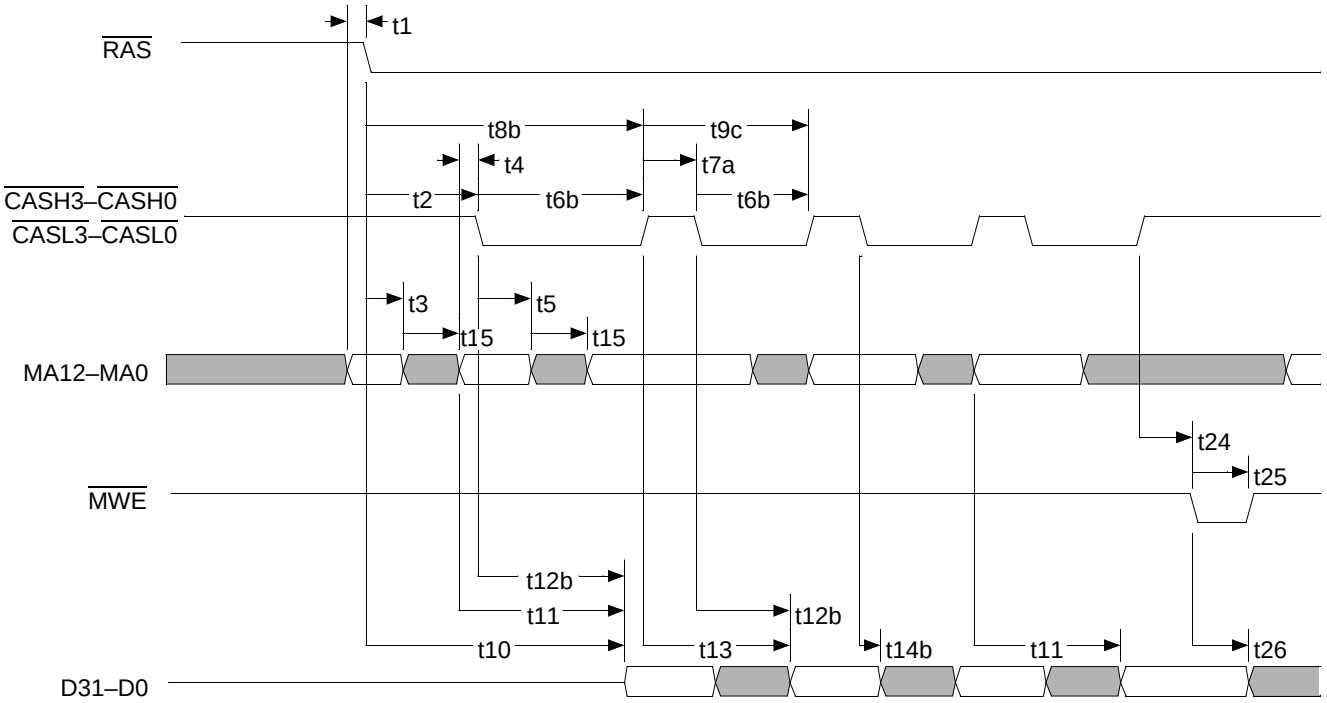


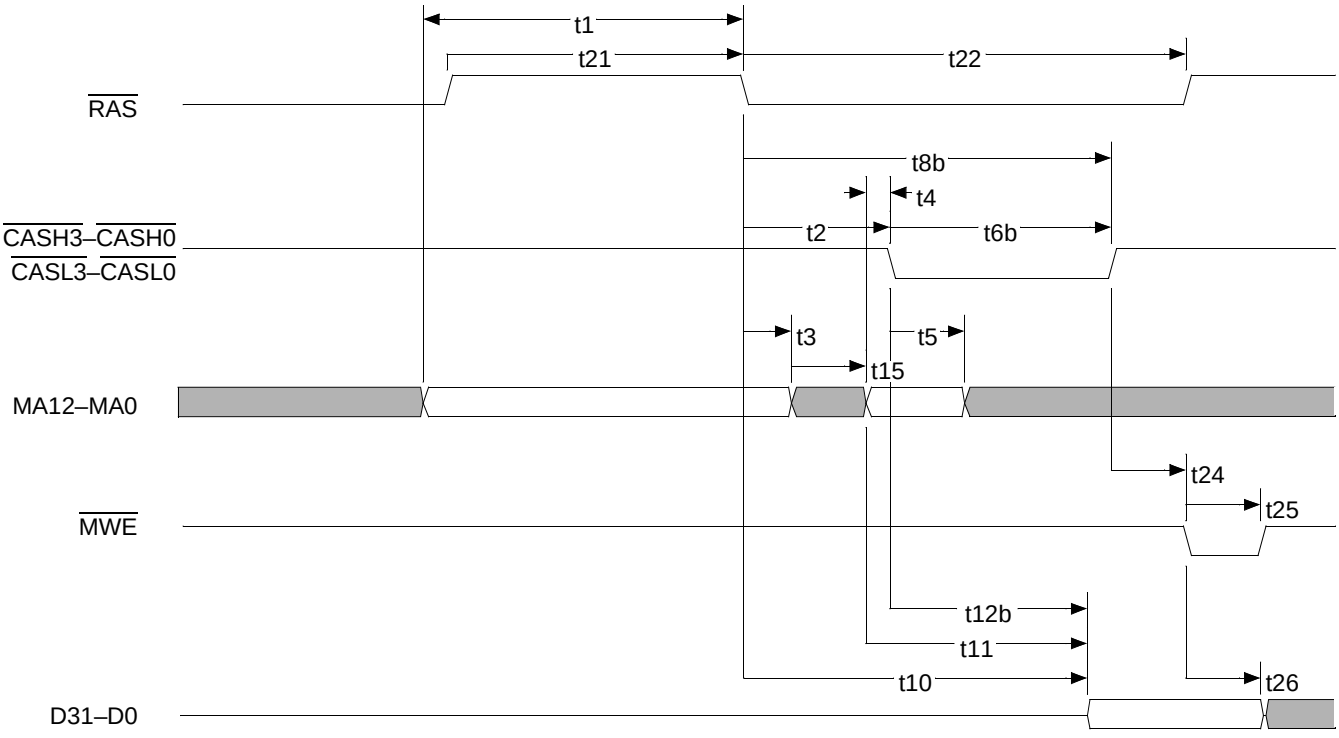
Figure 37. DRAM Page Miss Read, Non-Interleaved



Notes:

The EDO DRAM page hit write timing is similar to DRAM page hit write timing. See Figure 36 on page 101 for more information.

Figure 38. EDO DRAM Page Hit Read, Non-Interleaved



Notes:

The EDO DRAM page miss write timing is similar to DRAM page miss write timing. See Figure 36 on page 101 for more information.

Figure 39. EDO DRAM Page Miss Read, Non-Interleaved

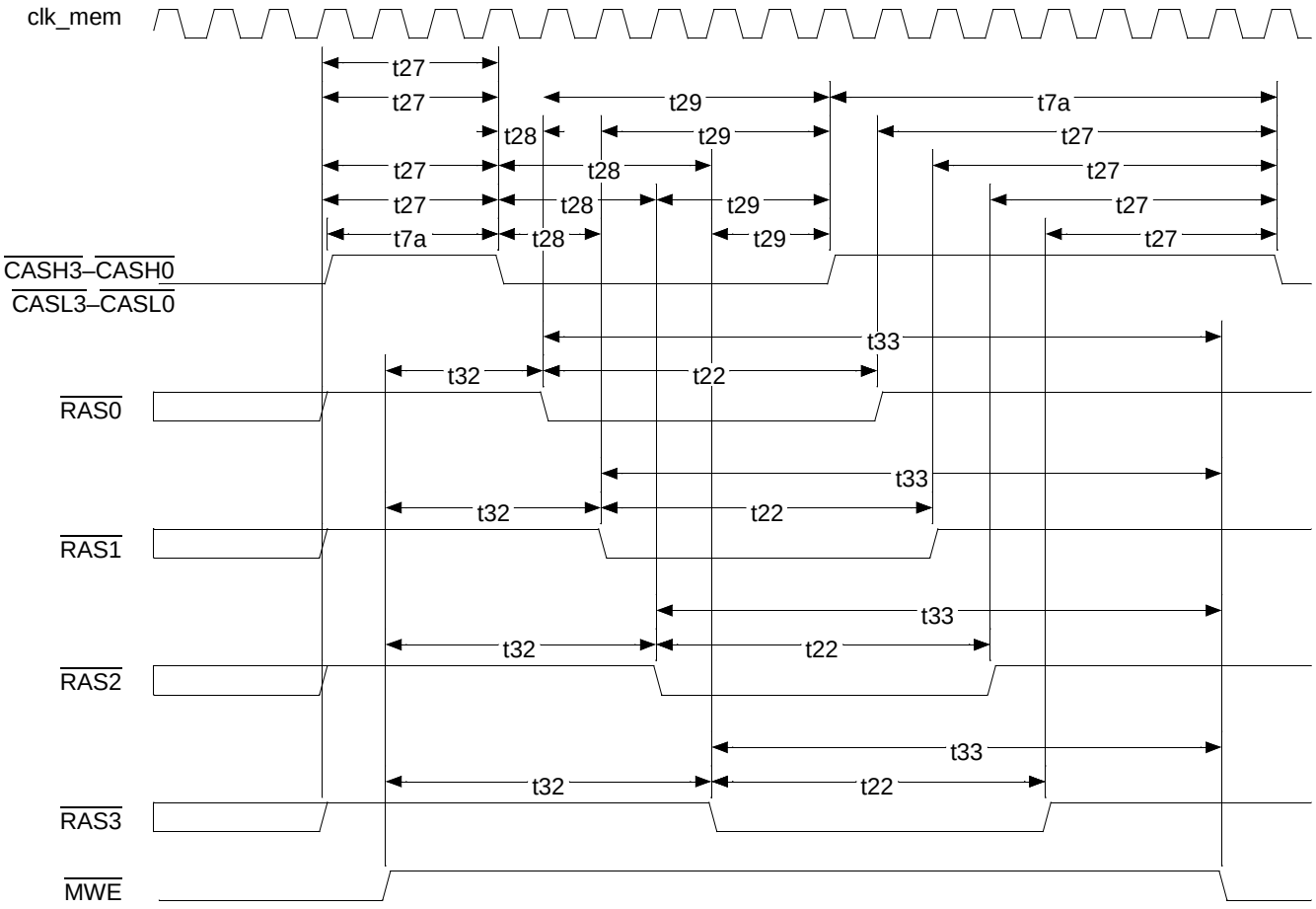
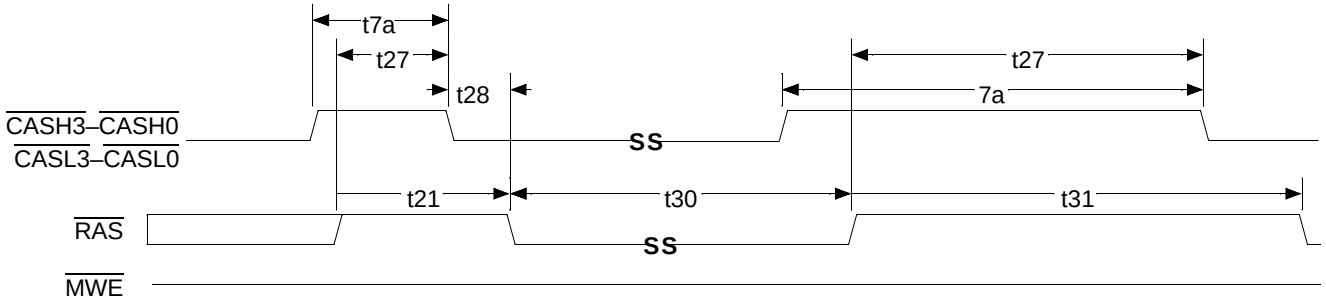


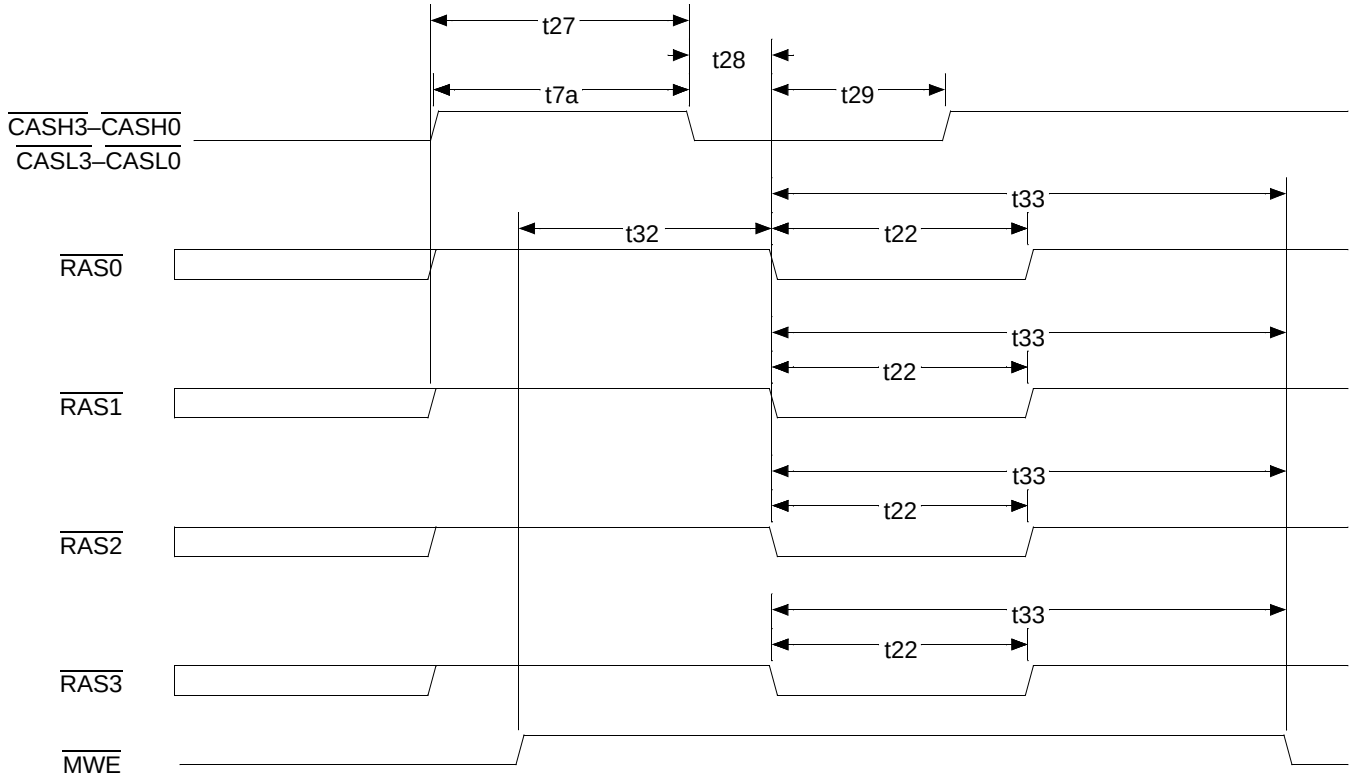
Figure 40. DRAM $\overline{\text{CAS}}$-Before-$\overline{\text{RAS}}$ Refresh



Notes:

Because the sequence shown above is performed when the microcontroller is in Suspend mode, the DRAMs must self-refresh. The $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ signals are held active (Low) for the entire time that the microcontroller is in Suspend mode. The timing diagram also shows a following cycle that brings $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ High again. The Low period of $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ can be of a long duration.

Figure 41. DRAM Self-Refresh



Notes:

The diagram above shows $\overline{RAS}$ and $\overline{CAS}$ behavior for an ÉlanSC400 or ÉlanSC410 microcontroller running at a frequency of 16 MHz or less. In this case, the $\overline{RAS}$ signals are not staggered and all are driven (Low) at the same time to consume less DRAM bandwidth for refresh activity (consumed due to a slower clock frequency).

Figure 42. DRAM Slow Refresh

Table 38. ISA Cycles

Symbol	Parameter Description	Notes	33-MHz External Bus		Unit
			Min	Max	
t1a	Setup, SA, $\overline{SBHE}$ stable to command assertion, 16-bit I/O, 8-bit I/O, Mem		120		ns
t1b	Setup, SA, $\overline{SBHE}$ stable to command assertion, 16-bit Mem		120		ns
t2a	Delay, $\overline{MCS16}$ stable from SA			102	ns
t2b	Delay, $\overline{IOCS16}$ stable from SA			122	ns
t3a	Pulse width, $\overline{IOW}$, 8-bit cycle		530		ns
t3b	Pulse width, $\overline{MEMW}$, 8-bit cycle		530		ns
t3c	Pulse width, $\overline{IOR}$, 8-bit cycle		530		ns
t3d	Pulse width, $\overline{MEMR}$, 8-bit cycle		530		ns
t3e	Pulse width, $\overline{IOW}$, 16-bit cycle		165		ns
t3f	Pulse width, $\overline{MEMW}$, 16-bit cycle		240		ns
t3g	Pulse width, $\overline{IOR}$, 16-bit cycle		165		ns
t3h	Pulse width, $\overline{MEMR}$, 16-bit cycle		240		ns
t4	SA, $\overline{SBHE}$ hold from command deassertion		53		ns
t5a	IOCHRDY delay from $\overline{IOR}$, $\overline{MEMR}$, $\overline{IOW}$, $\overline{MEMW}$ (8-bit)			378	ns
t5b	IOCHRDY delay from $\overline{IOR}$, $\overline{MEMR}$, $\overline{IOW}$, $\overline{MEMW}$ (16-bit)			66	ns
t6	$\overline{IOR}$, $\overline{MEMR}$, $\overline{IOW}$, $\overline{MEMW}$ delay from IOCHRDY		125		ns
t7a	$\overline{IOR}$, $\overline{MEMR}$, $\overline{IOW}$, $\overline{MEMW}$ high time (8-bit)		187		ns
t7b	$\overline{IOR}$, $\overline{MEMR}$, $\overline{IOW}$, $\overline{MEMW}$ high time (16-bit)		125		ns
t8	Delay, BALE rising from $\overline{IOR}$, $\overline{MEMR}$, $\overline{IOW}$, $\overline{MEMW}$ deassertion		46		ns
t9	IOCHRDY pulse width		120 ns	15.6 µs	
t11a	Setup, SD to write command assertion, 8-bit memory, I/O write and 16-bit I/O write		33		ns
t11b	Setup, SD to write command assertion, 16-bit memory write		-29		ns
t12	Hold, SD from write command deassertion		30		ns
t13a	Data access time, 8-bit read			489	ns
t13b	Data access time, 16-bit I/O read			132	ns
t13c	Data access time, 16-bit memory read			209	ns
t14	Hold, SD from read command deassertion		0		ns
t15	Setup, SA, $\overline{SBHE}$ stable to BALE falling edge		61		ns
t16	Pulse width, BALE		60		ns
t17	Setup, AEN high to $\overline{IOR}$/$\overline{IOW}$ assertion		145		ns
t19	Setup, SA, $\overline{SBHE}$ stable to command assertion		102		ns
t20	Hold, DRQ from DACK assertion		0		ns
t21	Setup, DACK assertion to I/O command assertion		145		ns
t22a	Setup, $\overline{IOR}$ assertion to $\overline{MEMW}$ command		235		ns
t22b	Setup, $\overline{MEMR}$ command assertion to $\overline{IOW}$ command		0		ns
t23	Delay, IOCHRDY assertion to command high		200		ns
t24	Delay, memory command to IOCHRDY deassertion			125	ns
t25	Hold, command off to DACK off		60		ns
t26	Hold, read command off from write command off		50		ns
t27	Hold, AEN from command off		60		ns

Table 38. ISA Cycles (Continued)

Symbol	Parameter Description	Notes	33-MHz External Bus		Unit
			Min	Max	
t29	Hold, SA, $\overline{\text{SBHE}}$ from read command		53		ns
t30	Setup, TC to read command deassertion		470		ns
t31	Hold, TC from read command deassertion		60		ns
t32a	Pulse width, I/O write command		400		ns
t32b	Pulse width, I/O read command		700		ns
t33a	Pulse width, memory read command		800		ns
t33b	Pulse width, memory write command		470		ns
t34	Delay, $\overline{\text{MEMR}}$ to valid data			272	ns
t35	Hold, SD from $\overline{\text{MEMR}}$ deassertion		11		ns
t36	Delay, $\overline{\text{IOR}}$ to valid data			241	ns
t37	Hold, SD from $\overline{\text{IOR}}$ deassertion		11		ns
t38	Setup, SD to $\overline{\text{MEMW}}$ assertion		-21		ns
t39	Setup, SD to $\overline{\text{IOW}}$ assertion		-214		ns
t41	Setup, $\overline{\text{DBUFRDL}}$/$\overline{\text{DBUFRDH}}$ to write command Low	1	45		ns
t42	Hold, $\overline{\text{DBUFRDL}}$/$\overline{\text{DBUFRDH}}$ from write command High	1	30		ns
t43	Setup, $\overline{\text{DBUFRDL}}$/$\overline{\text{DBUFRDH}}$ to read command Low	1	0		ns
t44	Hold, $\overline{\text{DBUFRDL}}$/$\overline{\text{DBUFRDH}}$ from read command High	1	10		ns
t45	Setup, $\overline{\text{DBUFOE}}$ Low to write command Low	1	45		ns
t46	Hold, $\overline{\text{DBUFOE}}$ from write command High	1	30		ns
t47	Setup, $\overline{\text{DBUFOE}}$ Low to read command Low	1	0		ns
t48	Hold, $\overline{\text{DBUFOE}}$ from read command High	1	10		ns
t49	Setup, $\overline{\text{DBUFRDL}}$/$\overline{\text{DBUFRDH}}$ to mem read command Low, DMA	1	0		ns
t50	Hold, $\overline{\text{DBUFRDL}}$/$\overline{\text{DBUFRDH}}$ from mem read command High, DMA	1	10		ns
t51	Setup, $\overline{\text{DBUFOE}}$ Low to mem read command Low, DMA	1	0		ns
t52	Hold, $\overline{\text{DBUFOE}}$ from mem read command High, DMA	1	10		ns
t53	Setup, $\overline{\text{DBUFRDL}}$/$\overline{\text{DBUFRDH}}$ to I/O read command Low, DMA	1	0		ns
t54	Setup, $\overline{\text{DBUFOE}}$ Low to I/O read command Low, DMA	1	0		ns
t55	Hold, $\overline{\text{DBUFOE}}$ from I/O read command High, DMA	1	10		ns
t56	Hold, $\overline{\text{DBUFRDL}}$/$\overline{\text{DBUFRDH}}$ from I/O read command High, DMA	1	10		ns

Notes:

1. These parameters are applicable only when an external data transceiver is used to isolate the SD bus.

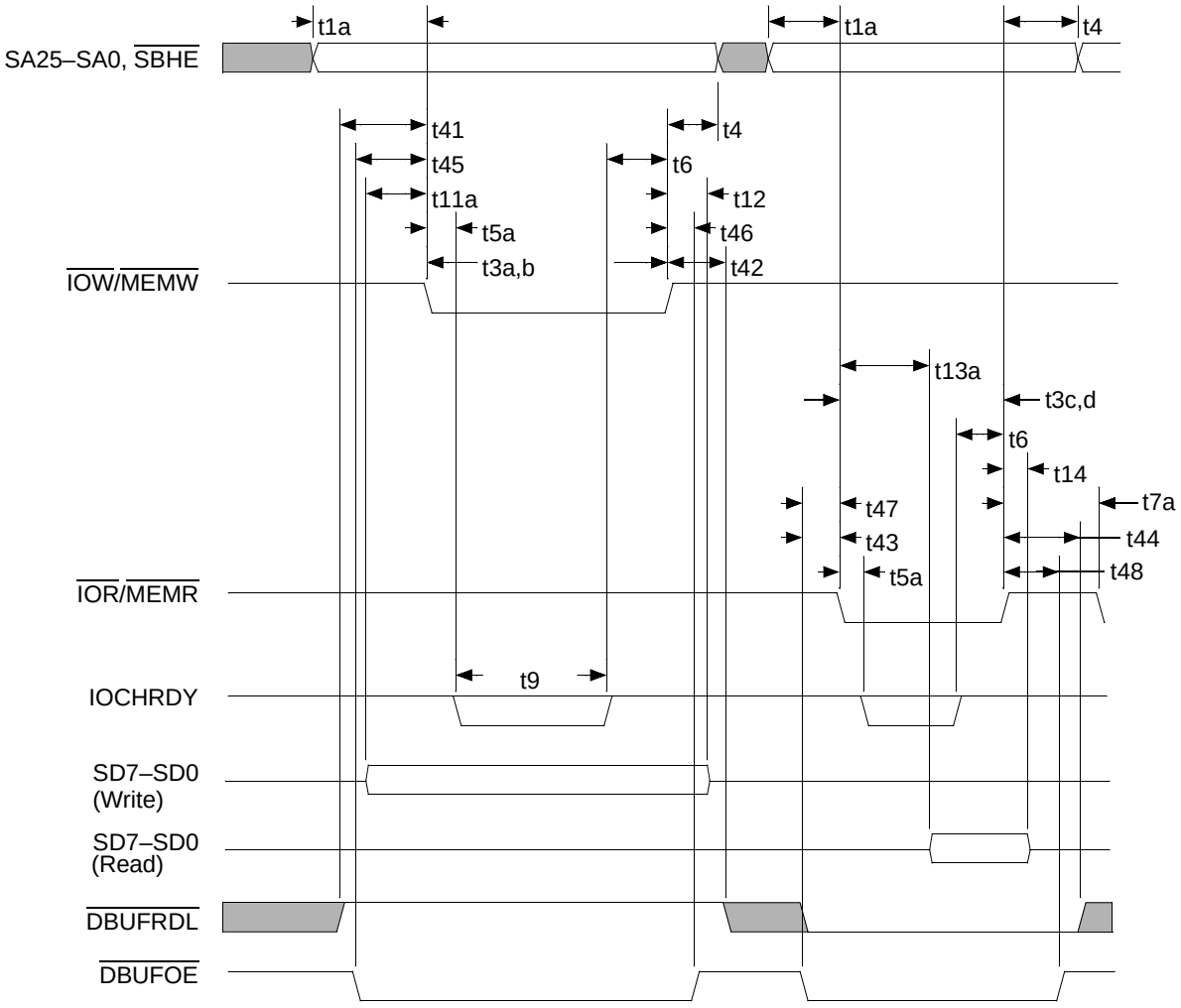


Figure 43. 8-Bit ISA Bus Cycles

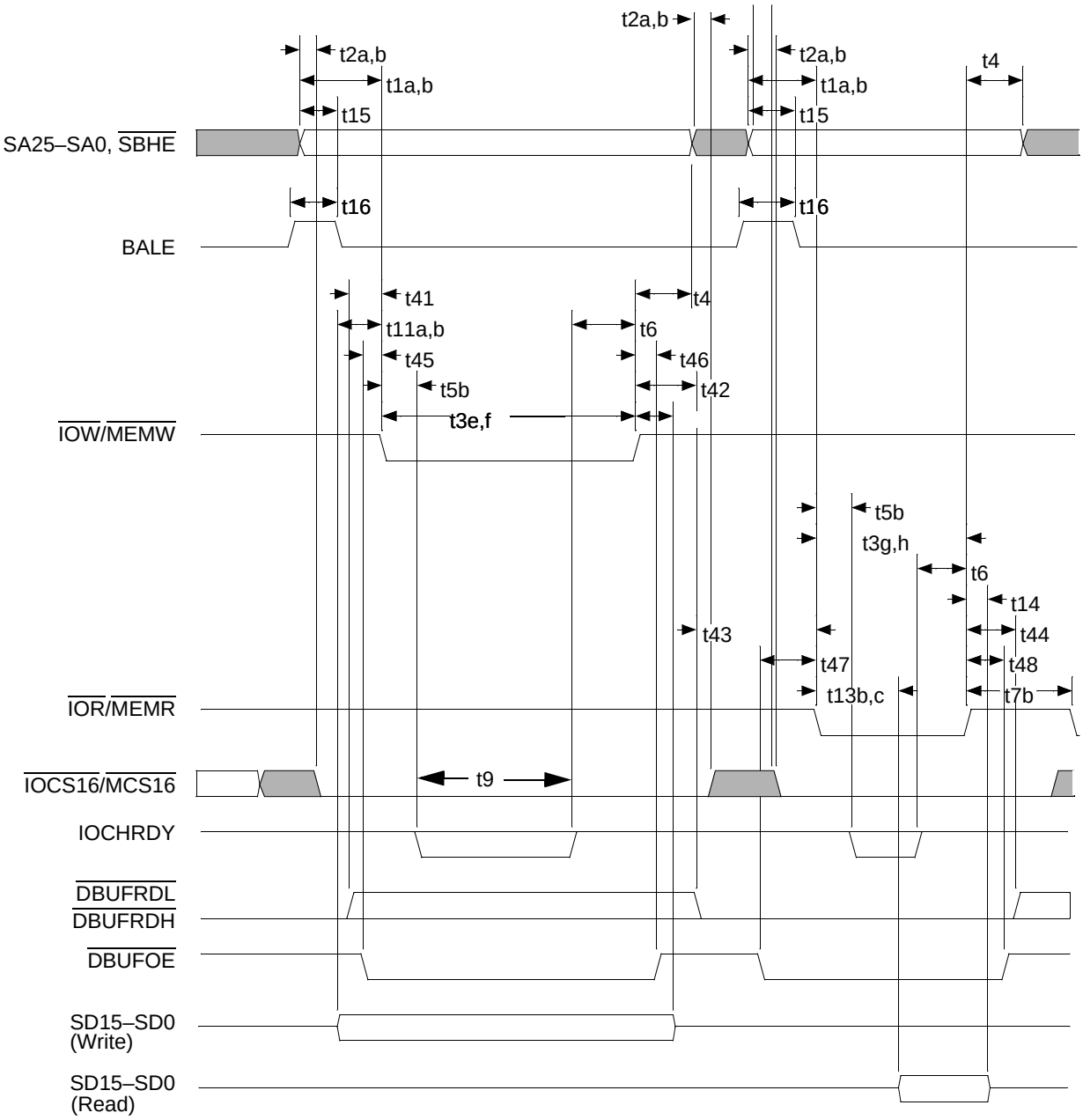


Figure 44. 16-Bit ISA Bus Cycles

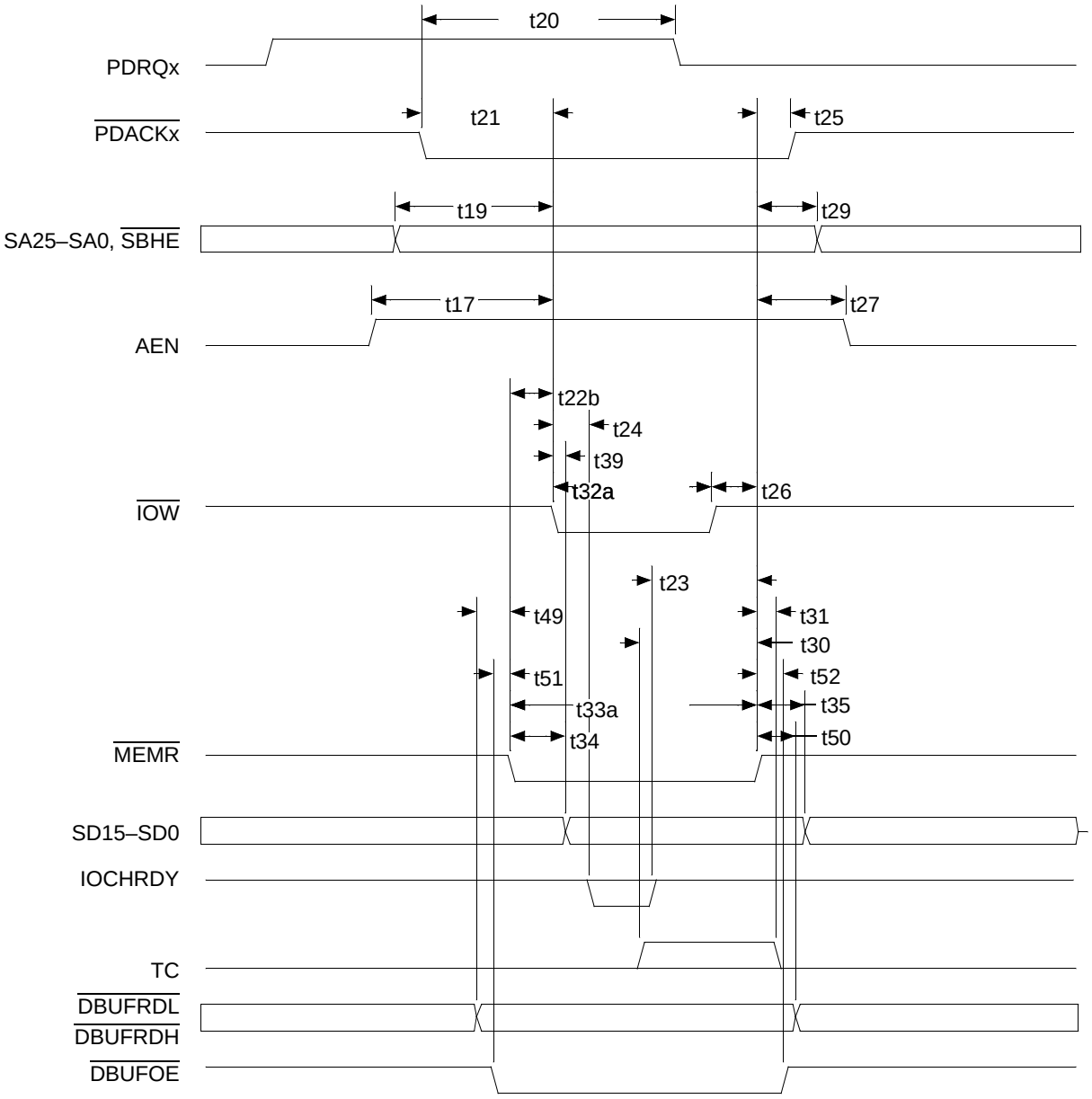


Figure 45. ISA DMA Read Cycle

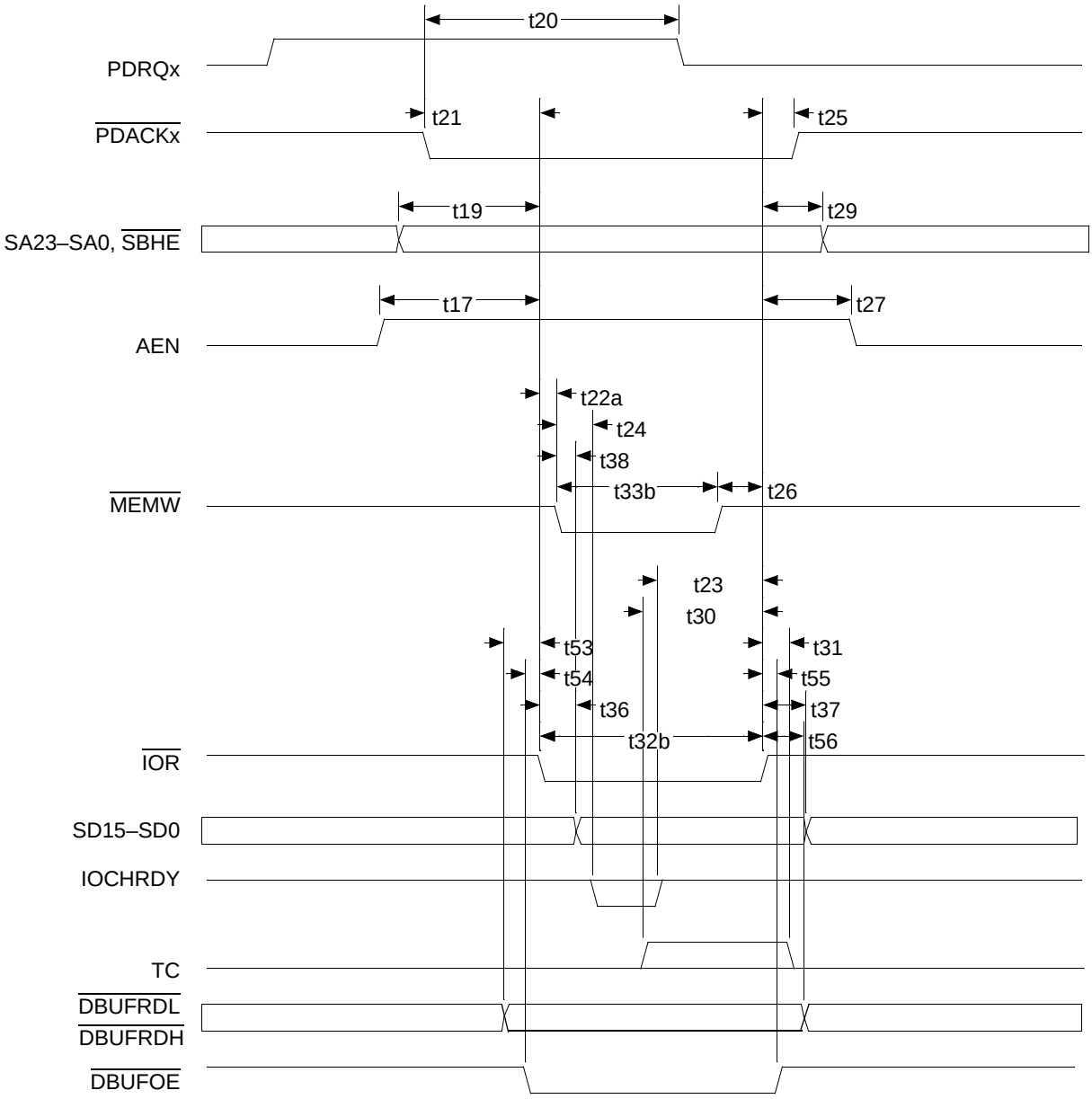


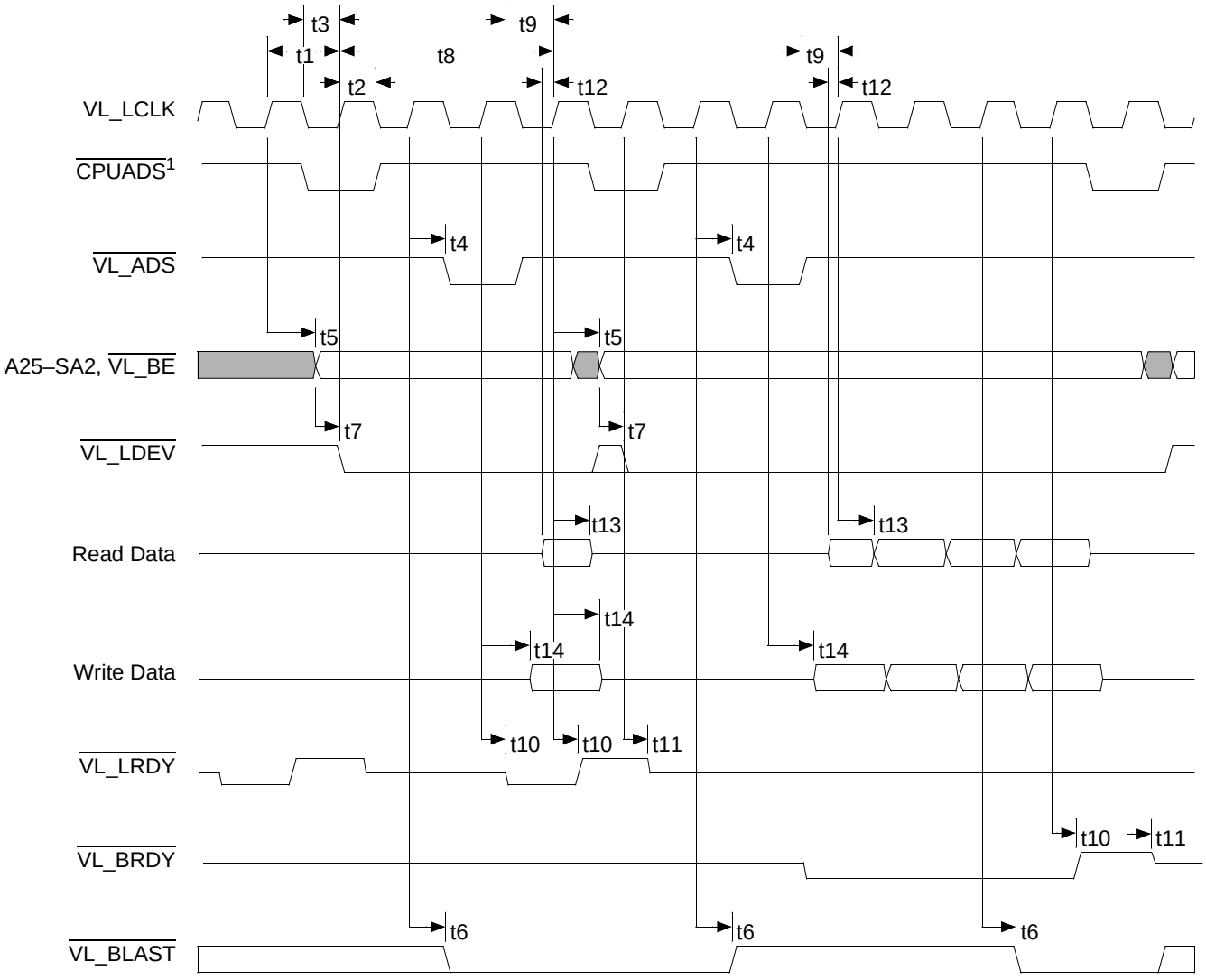
Figure 46. ISA DMA Write Cycle

Table 39. VESA Local Bus Cycles

Symbol	Parameter Description	33-MHz External Bus		Unit
		Min	Max	
t1	VL_LCLK period	27		ns
t2	VL_LCLK pulse High	14		ns
t3	VL_LCLK pulse Low	14		ns
t4	$\overline{\text{VL\_ADS}}$ delay from VL_LCLK	3	18	ns
t5	SA25–SA2, $\overline{\text{VL\_BE3}}$–$\overline{\text{VL\_BE0}}$, VL_M/$\overline{\text{IO}}$, VL_W/$\overline{\text{R}}$, VL_D/$\overline{\text{C}}$ delay from VL_LCLK	3	18	ns
t6	$\overline{\text{VL\_BLAST}}$ valid from VL_LCLK	3	18	ns
t7	$\overline{\text{VL\_LDEV}}$ valid from SA25–SA2, $\overline{\text{VL\_BE3}}$–$\overline{\text{VL\_BE0}}$, VL_M/$\overline{\text{IO}}$, VL_W/$\overline{\text{R}}$, VL_D/$\overline{\text{C}}$		20	ns
t8[1]	$\overline{\text{VL\_LDEV}}$ setup to VL_LCLK	15		ns
t9	$\overline{\text{VL\_LRDY}}$, $\overline{\text{VL\_BRDY}}$ setup to VL_LCLK	12		ns
t10	$\overline{\text{VL\_LRDY}}$, $\overline{\text{VL\_BRDY}}$ (VL-Bus target is driver) hold from VL_LCLK	0		ns
t11	$\overline{\text{VL\_LRDY}}$ (VL-Bus target is driver) three stated from VL_LCLK		0	ns
t12	Read data setup to VL_LCLK	5		ns
t13	Read data hold from VL_CLK	0		ns
t14	Write data delay from VL_CLK	3	18	ns

Notes:

1. LDEV is checked on the following rising edge of the CPU clock (not shown, up to 100 MHz) from the assertion of ADS. ADS can assert a minimum of 20 ns after address change.



Notes:

1. This signal is shown as a timing reference only. It is not available as a pin on the ÉlanSC400 microcontroller.

Figure 47. VESA Local Bus Cycles

Table 40. Parallel Port Cycles[1]

Symbol	Parameter Description	Notes	33-MHz External Bus		Unit
			Min	Max	
t1	$\overline{\text{PPDWE}}$ delay from $\overline{\text{IOW}}$	2	2	20	ns
t2	$\overline{\text{PPOEN}}$ delay from $\overline{\text{IOW}}$		2	20	ns
t3	$\overline{\text{STRB}}$ delay from $\overline{\text{IOW}}$		2	20	ns
t4	$\overline{\text{SLCTIN}}$, $\overline{\text{AFDT}}$ valid from $\overline{\text{IOW}}$	3	2	20	ns
t5	SD setup to $\overline{\text{IOW}}$		50		ns
t6	SD hold from $\overline{\text{IOW}}$		50		ns
t7	BUSY asserted from $\overline{\text{IOW}}$ asserted	4		300	ns
t8	$\overline{\text{IOW}}$ deasserted from BUSY deasserted	4	100		ns
t9	$\overline{\text{IOW}}$ pulse width		450		ns
t10	$\overline{\text{SLCTIN}}$, $\overline{\text{AFDT}}$ recovery		1000		ns
t11	$\overline{\text{DBUFOE}}$ setup to $\overline{\text{IOW}}$	5	20		ns
t12	$\overline{\text{DBUFOE}}$ hold from $\overline{\text{IOW}}$	5	20		ns
t13	$\overline{\text{PPDWE}}$ delay from $\overline{\text{IOR}}$	2	2	20	ns
t14	$\overline{\text{SLCTIN}}$, $\overline{\text{AFDT}}$ valid from $\overline{\text{IOR}}$	3	2	20	ns
t15	SD setup to $\overline{\text{IOR}}$ deasserted		20		ns
t16	SD hold from $\overline{\text{IOR}}$		0		ns
t17	BUSY asserted from $\overline{\text{IOR}}$ asserted	4		300	ns
t18	$\overline{\text{IOR}}$ deasserted from BUSY deasserted	4	100		ns
t19	$\overline{\text{IOR}}$ pulse width		450		ns
t20	$\overline{\text{DBUFOE}}$, $\overline{\text{DBUFRDL}}$ setup to $\overline{\text{IOR}}$	5	0		ns
t21	$\overline{\text{DBUFOE}}$, $\overline{\text{DBUFRDL}}$ hold from $\overline{\text{IOR}}$	5	10		ns

Notes:

1. The signal names used in Figure 48 and Figure 49 are the PC/AT Compatible and Bidirectional mode signal names.

2. During EPP mode and Bidirectional mode, $\overline{\text{PPDWE}}$ acts as the parallel port chip select and is asserted for both reads and writes. For PC/AT Compatible mode, $\overline{\text{PPDWE}}$ will be asserted only for parallel port write cycles.

3. These timings are only valid for EPP mode.

4. BUSY is asserted to add wait states to the parallel port access.

5. $\overline{\text{DBUFOE}}$ and $\overline{\text{DBUFRDL}}$ may be required when using the VESA local bus interface or a x32 DRAM interface.

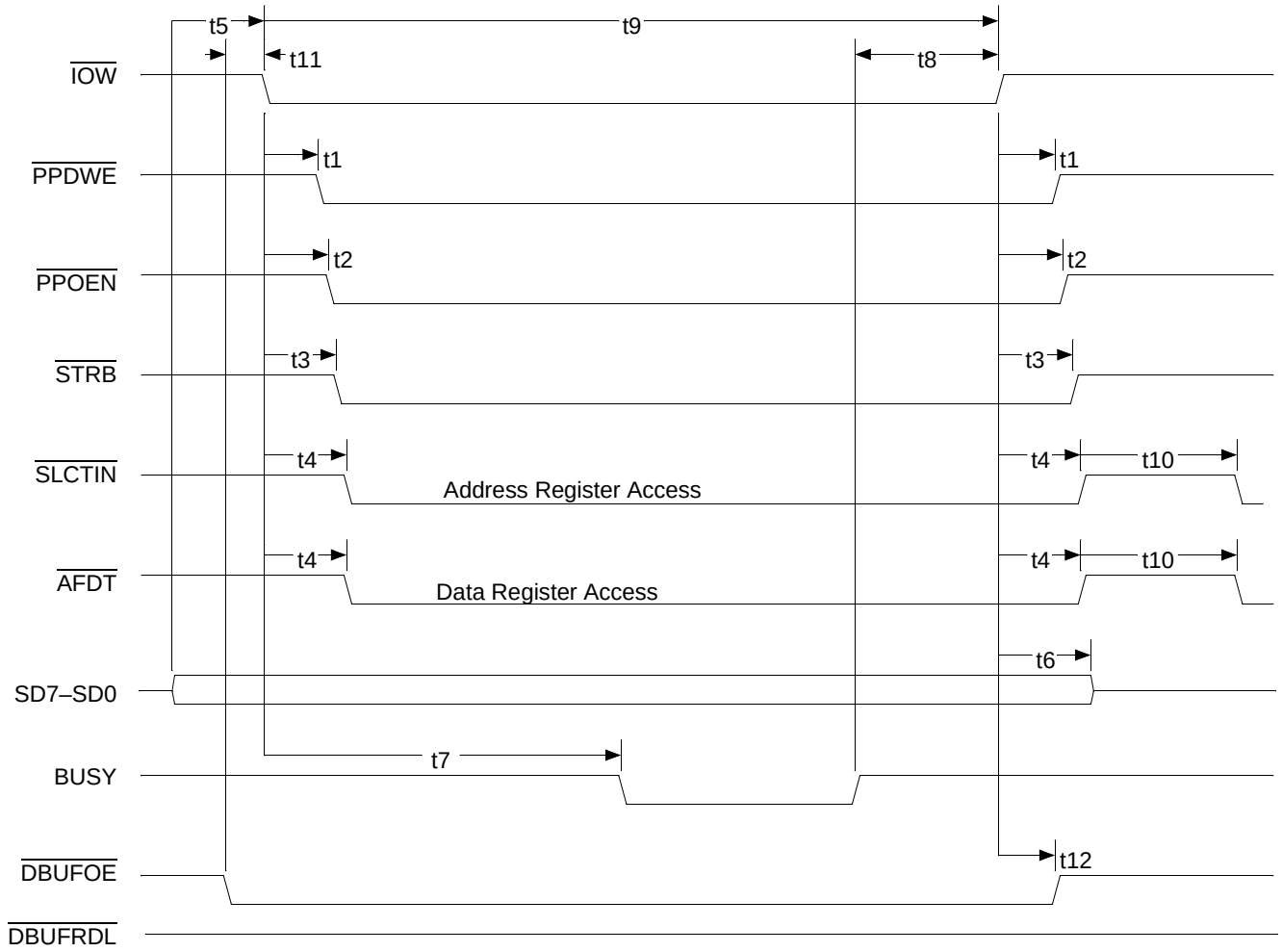


Figure 48. EPP Parallel Port Write Cycle

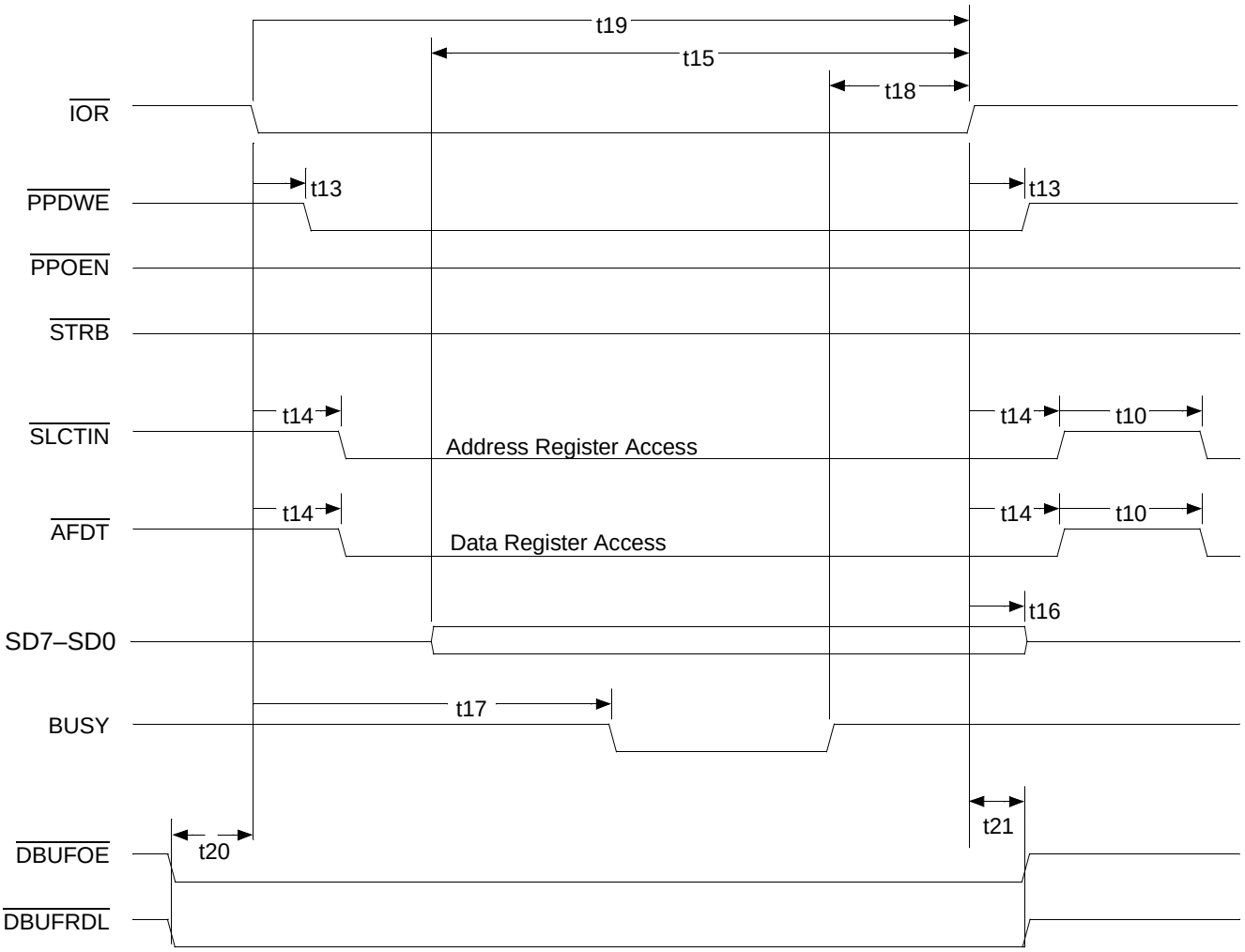
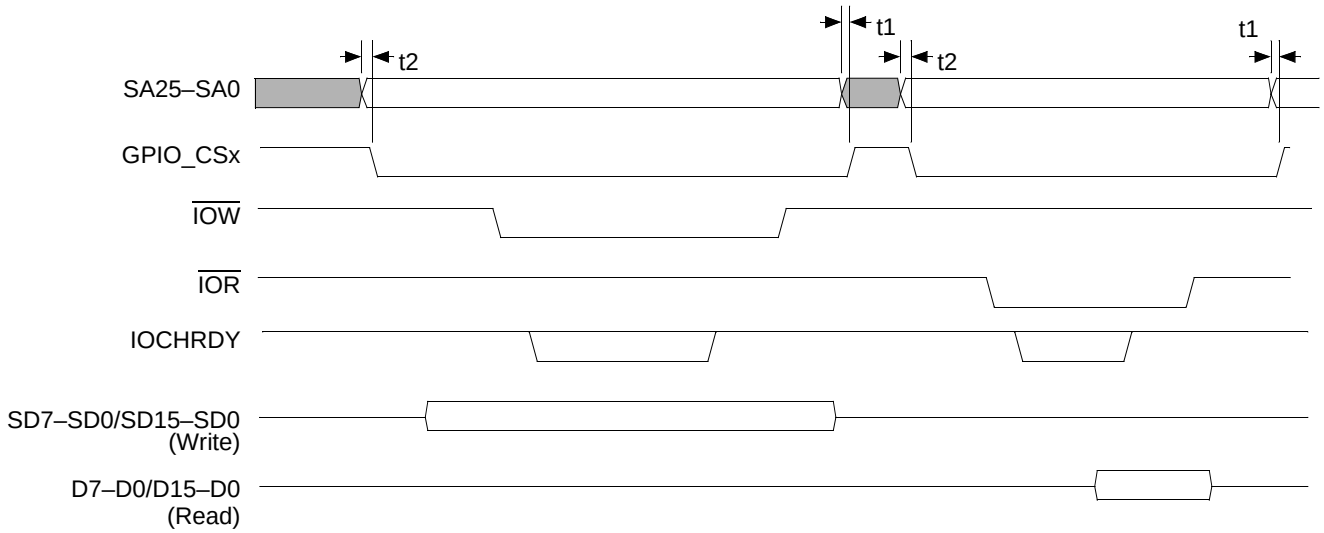


Figure 49. EPP Parallel Port Read Cycle

Table 41. General-Purpose Input/Output Cycles

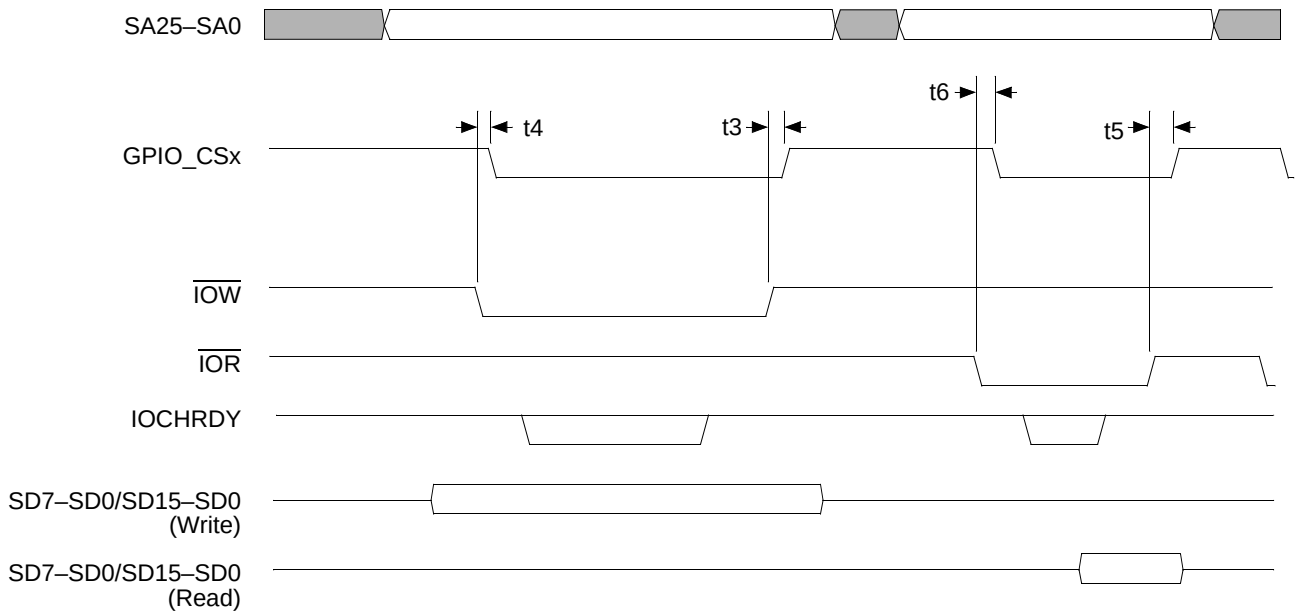
Symbol	Parameter Description	33-MHz External Bus		Unit
		Min	Max	
t1	SA stable to GPIO_CSx rising edge		10	ns
t2	SA stable to GPIO_CSx falling edge		10	ns
t3	$\overline{IOW}$ rising edge to GPIO_CSx rising edge		5	ns
t4	$\overline{IOW}$ falling edge to GPIO_CSx falling edge		5	ns
t5	$\overline{IOR}$ rising edge to GPIO_CSx rising edge		5	ns
t6	$\overline{IOR}$ falling edge to GPIO_CSx falling edge		5	ns
t7	SA stable to GPIO_CSx ($\overline{8042CS}$) falling edge		10	ns
t8	SA stable to GPIO_CSx ($\overline{8042CS}$) rising edge		10	ns
t9	SA stable to GPIO_CSx ($\overline{MEMCS}$) falling edge		10	ns
t10	SA stable to GPIO_CSx ($\overline{MEMCS}$) rising edge		10	ns
t11	$\overline{MEMW}$ rising edge to GPIO_CSx rising edge		5	ns
t12	$\overline{MEMW}$ falling edge to GPIO_CSx falling edge		5	ns
t13	$\overline{MEMR}$ rising edge to GPIO_CSx rising edge		5	ns
t14	$\overline{MEMR}$ falling edge to GPIO_CSx falling edge		5	ns



Notes:

See the ISA bus section on page 105 for detailed timings between these signals.

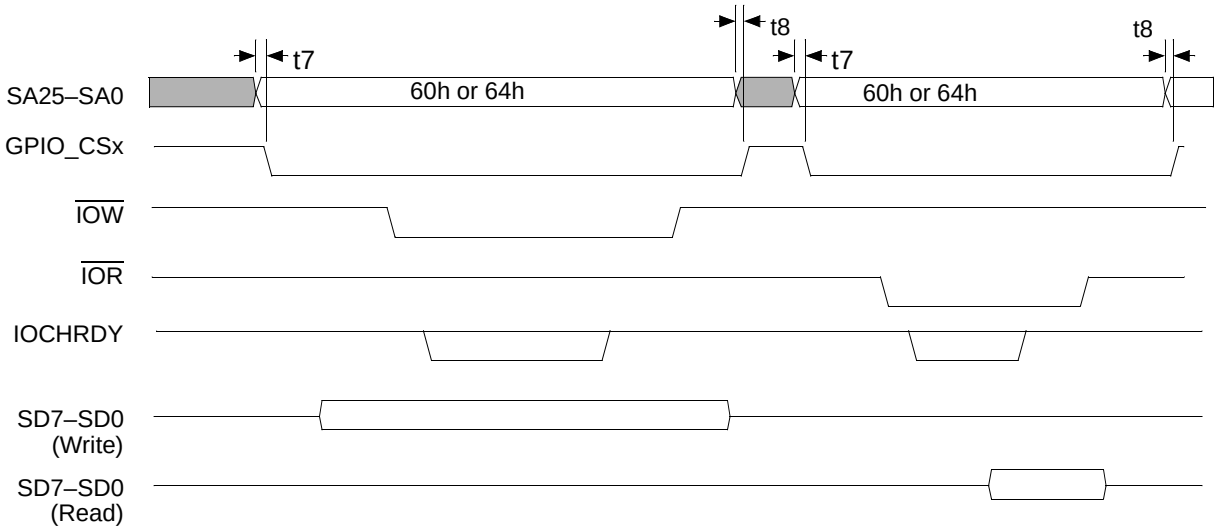
Figure 50. I/O Decode (R/W), Address Decode Only



Notes:

See the ISA bus section on page 105 for detailed timings between these signals.

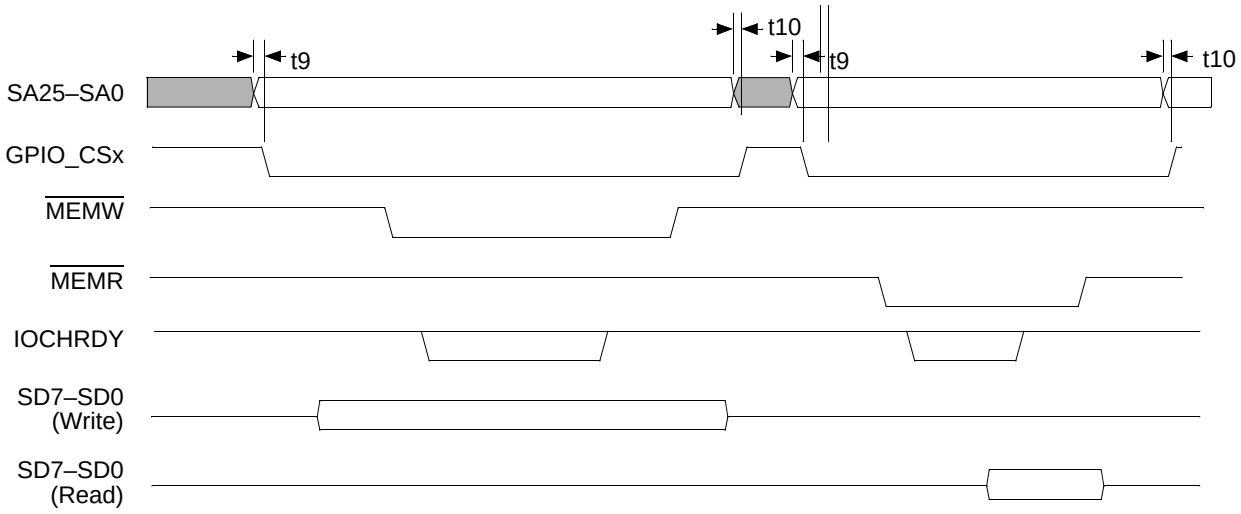
Figure 51. I/O Decode (R/W), Command Qualified



Notes:

See the ISA bus section on page 105 for detailed timings between these signals.

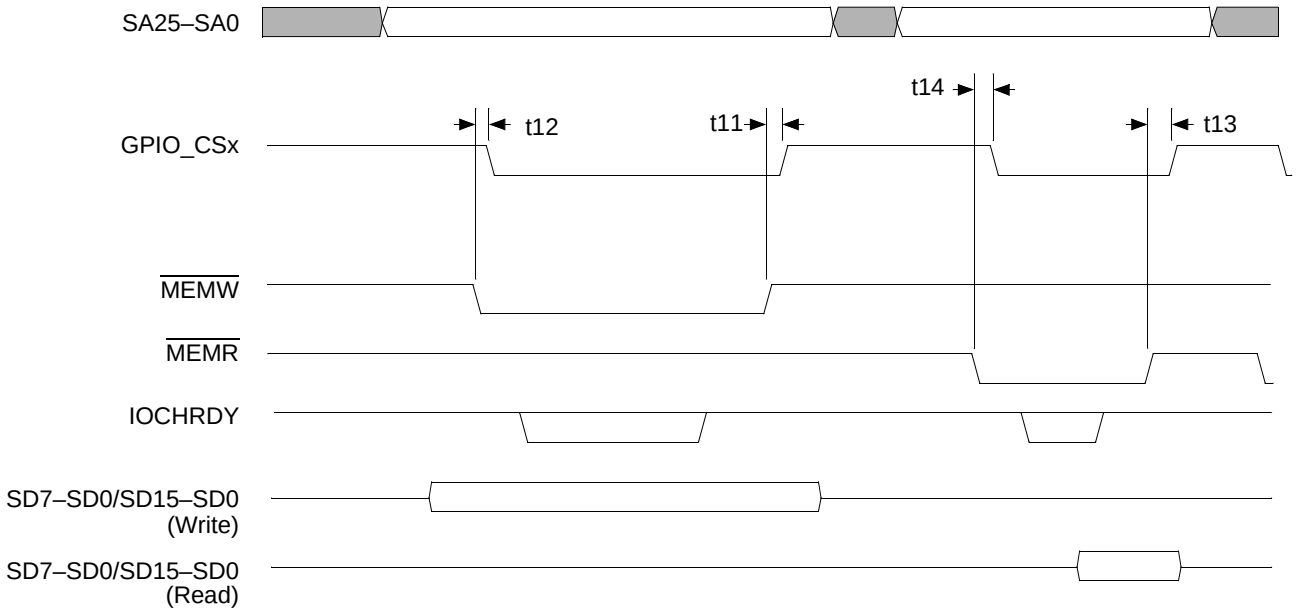
Figure 52. I/O Decode (R/W), GPIO_CSx as $\overline{8042CS}$ Timing



Notes:

See the ISA bus section on page 105 for detailed timings between these signals.

Figure 53. Memory $\overline{CS}$ Decode (R/W), Address Decode Only



Notes:

See the ISA bus section on page 105 for detailed timings between these signals.

Figure 54. Memory $\overline{CS}$ Decode (R/W), Command Qualified

Table 42. PC Card Cycles—ÉlanSC400 Microcontroller Only

Symbol	Parameter Description	Notes	33-MHz External Bus		Unit
			Min	Max	
t1	$\overline{\text{REG\_x}}$, SA setup to command active	1, 2	ST-10		ns
t2	Command pulse width	1, 3	CT-10		ns
t3	SA hold and write data valid from command inactive	1, 4	RT-10		ns
t4	$\overline{\text{WAIT\_AB}}$ Active from command active	1, 3		(C-2)•T-10	ns
t5	Command hold from $\overline{\text{WAIT\_AB}}$ inactive	1	2T		ns
t6	SD setup before read command inactive	1	2T+10		ns
t7	SD valid from read command inactive		0		ns
t8	SD valid from $\overline{\text{WAIT\_AB}}$ inactive	1	T+10		ns
t9	$\overline{\text{IOIS16}}$ setup before command inactive	1	3T+10		ns
t10	$\overline{\text{MCEH\_x}}$ delay from $\overline{\text{IOIS16}}$ active	1	T-10		ns
t11	$\overline{\text{IOIS16}}$ delay from valid SA			35	ns
t12	Setup, DACK assertion to DMA I/O command active		145		ns
t13a	Pulse width, DMA I/O write command		220		ns
t13b	Pulse width, DMA I/O read command		700		ns
t14	Hold, DMA I/O command inactive to DACK inactive		60		ns
t15	DMA I/O command setup to $\overline{\text{TC}}$ active		15		ns
t16	$\overline{\text{TC}}$ pulse width		62		ns
t17a	SD setup to DMA $\overline{\text{IOW}}$ active		-241		ns
t17b	SD valid delay from DMA $\overline{\text{IOR}}$ active			100	ns
t18a	SD hold from DMA $\overline{\text{IOW}}$ inactive		0		ns
t18b	SD hold from DMA $\overline{\text{IOR}}$ inactive		0		ns

Notes:

1. *T is the nominal period of the selected clock: in Standard mode, this is the 125-ns ISA bus clock; in Enhanced mode, it is the 30-ns local bus clock.*
2. *S determines the setup time as programmed into the Setup Timing Register selected from one of four timing sets. Its value can be programmed to a range of 1 to 4096 • 63.*
3. *C determines the command active time as programmed into the Command Timing Register selected from one of four timing sets. Its value can be programmed to a range of 1 to 4096 • 63.*
4. *R determines the recovery time as programmed into the Recovery Timing Register selected from one of four timing sets. Its value can be programmed to a range of 1 to 4096 • 63.*

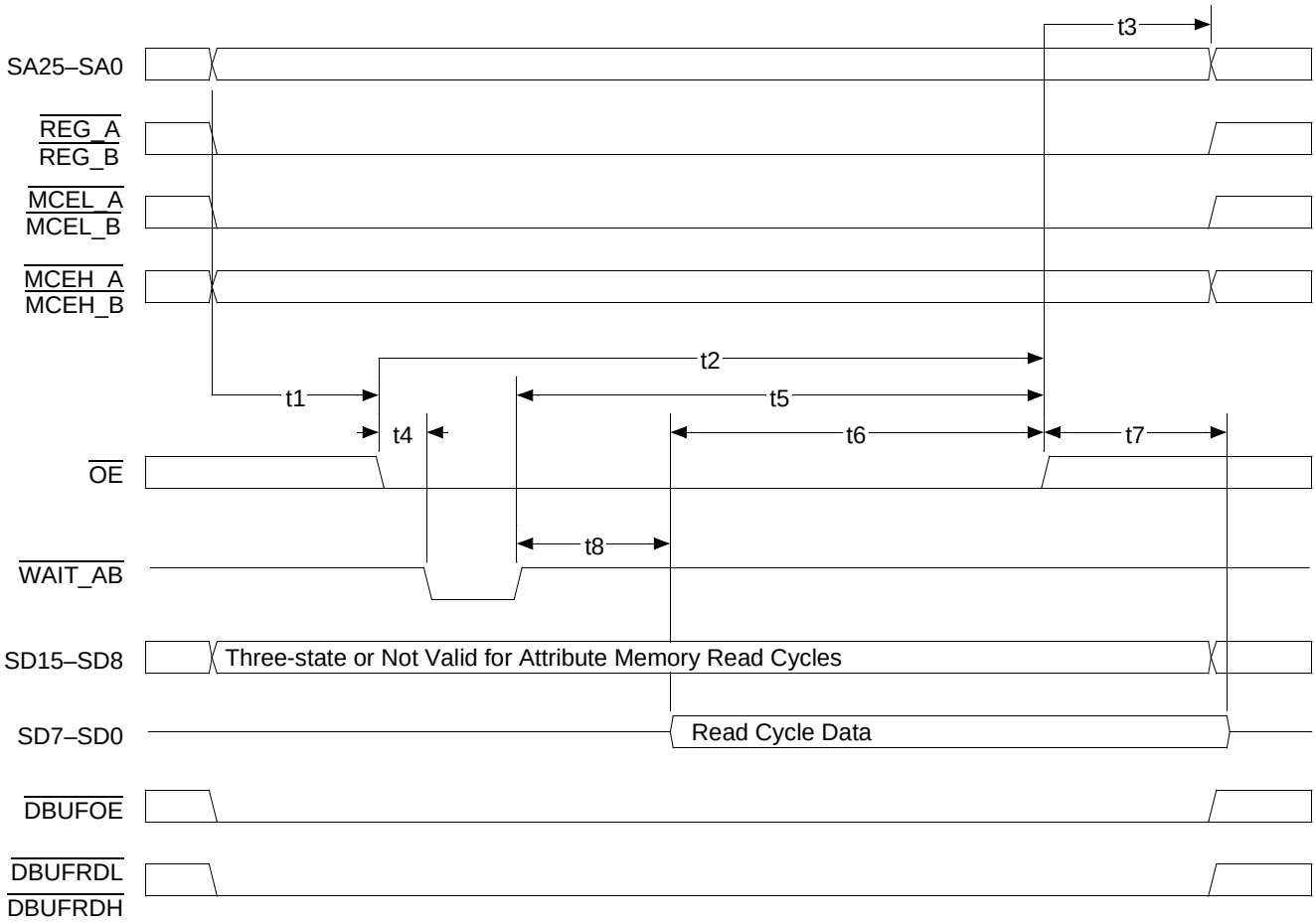


Figure 55. PC Card Attribute Memory Read Cycle (ÉlanSC400 Microcontroller Only)

Table 43. PC Card Attribute Memory Read Function (ÉlanSC400 Microcontroller Only)

Mode	$\overline{\text{REG\_x}}$	$\overline{\text{MCEH\_x}}$	$\overline{\text{MCEL\_x}}$	SA0	$\overline{\text{OE}}$	$\overline{\text{WE}}$	SD15–SD8	SD7–SD0
Byte Access	L L	H H	L L	L H	L L	H H	Three-state Three-state	Even byte Not valid
Word Access	L	L	L	Indeterminate	L	H	Not valid	Even byte
Odd-Byte-Only Access	L	L	H	Indeterminate	L	H	Not valid	Three-state

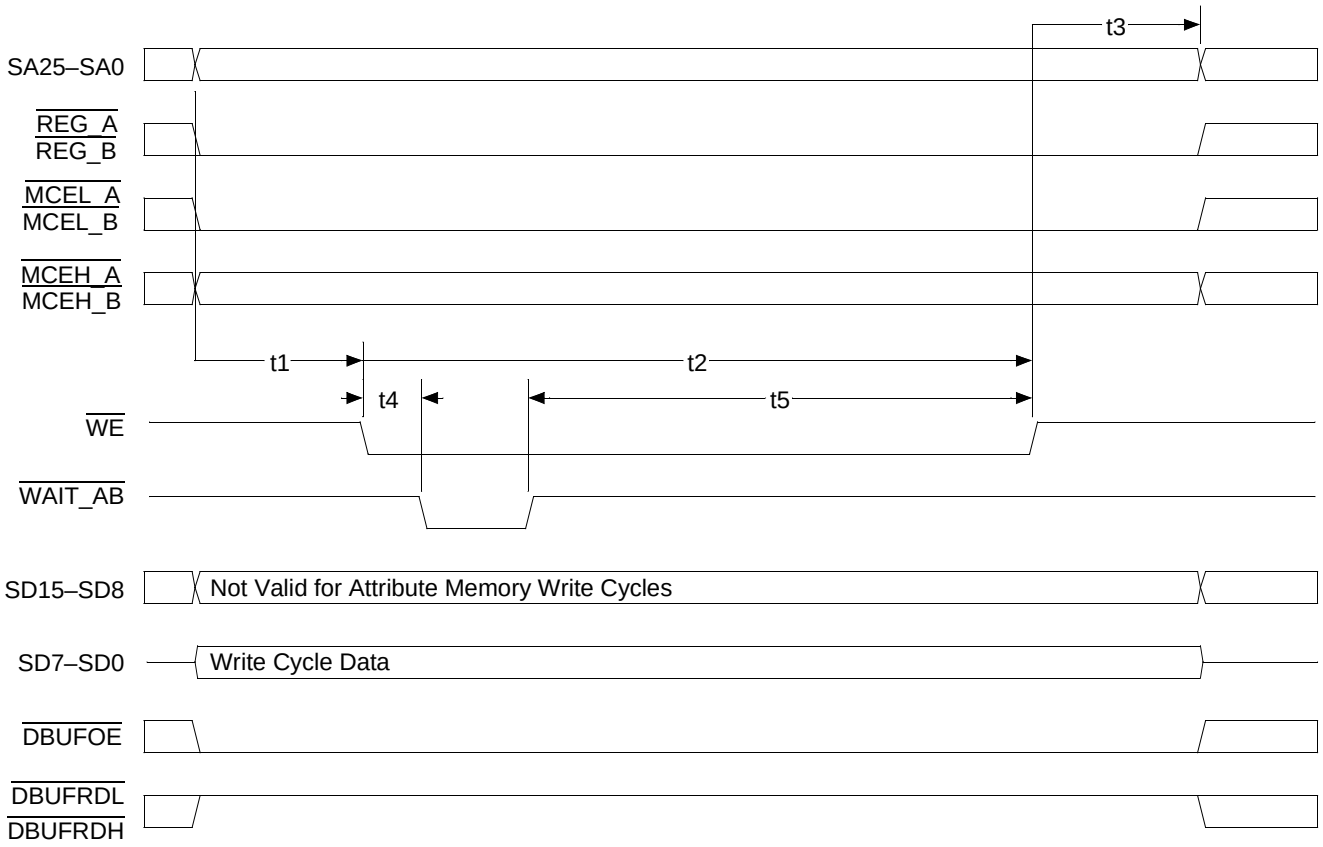


Figure 56. PC Card Attribute Memory Write Cycle (ÉlanSC400 Microcontroller Only)

Table 44. PC Card Attribute Memory Write Function (ÉlanSC400 Microcontroller Only)

Mode	$\overline{\text{REG\_x}}$	$\overline{\text{MCEH\_x}}$	$\overline{\text{MCEL\_x}}$	SA0	$\overline{\text{OE}}$	$\overline{\text{WE}}$	SD15–SD8	SD7–SD0
Byte Access	L L	H H	L L	L H	H H	L L	Indeterminate Indeterminate	Even byte Indeterminate
Word Access	L	L	L	Indeterminate	H	L	Indeterminate	Even byte
Odd-Byte-Only Access	L	L	H	Indeterminate	H	L	Indeterminate	Indeterminate

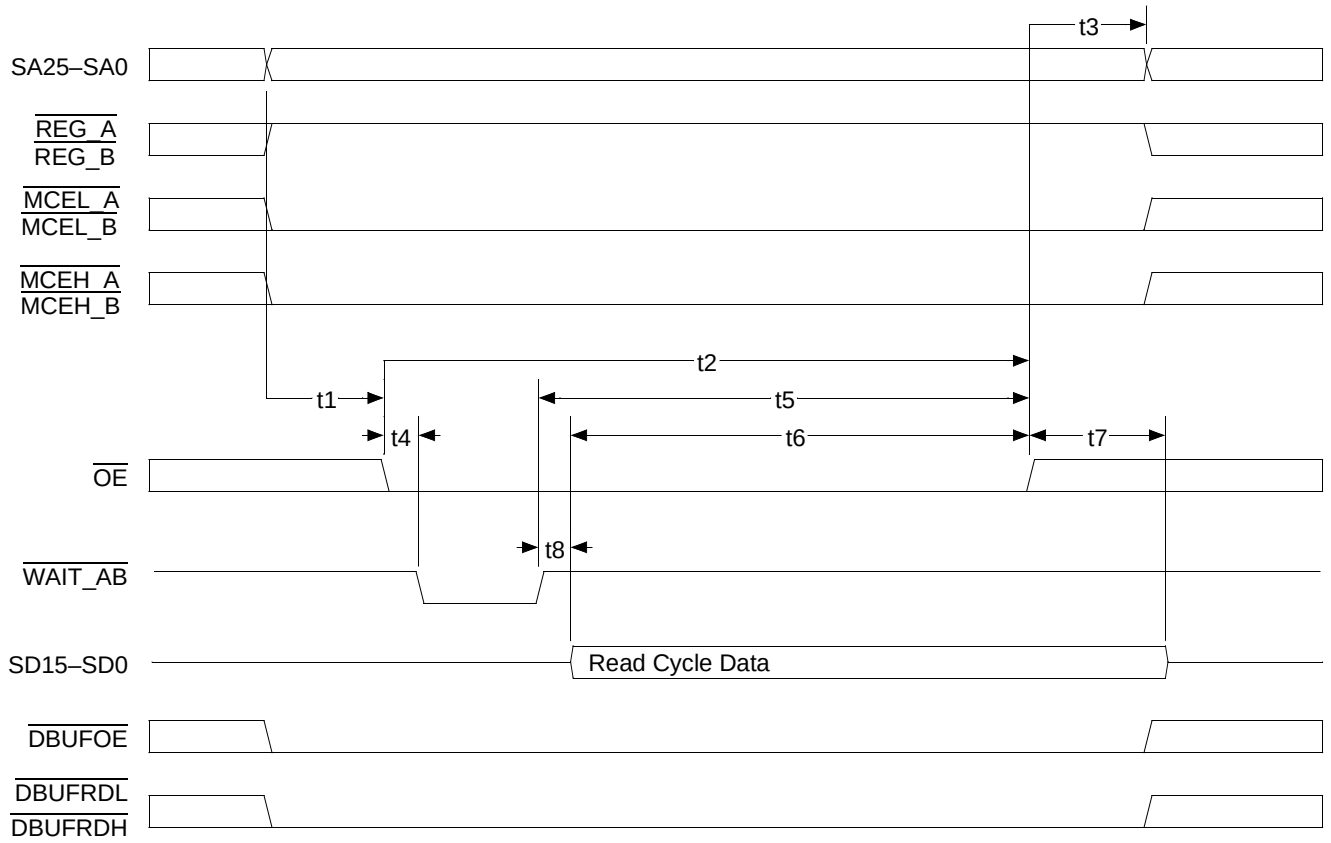


Figure 57. PC Card Common Memory Read Cycle (ÉlanSC400 Microcontroller Only)

Table 45. PC Card Common Memory Read Function (ÉlanSC400 Microcontroller Only)

Mode	$\overline{\text{REG\_x}}$	$\overline{\text{MCEH\_x}}$	$\overline{\text{MCEL\_x}}$	SA0	$\overline{\text{OE}}$	$\overline{\text{WE}}$	SD15–SD8	SD7–SD0
Byte Access	H H	H H	L L	L H	L L	H H	Three-state Three-state	Even byte Odd byte
Word Access	H	L	L	Indeterminate	L	H	Odd byte	Even byte
Odd-Byte-Only Access	H	L	H	Indeterminate	L	H	Odd byte	Three-state

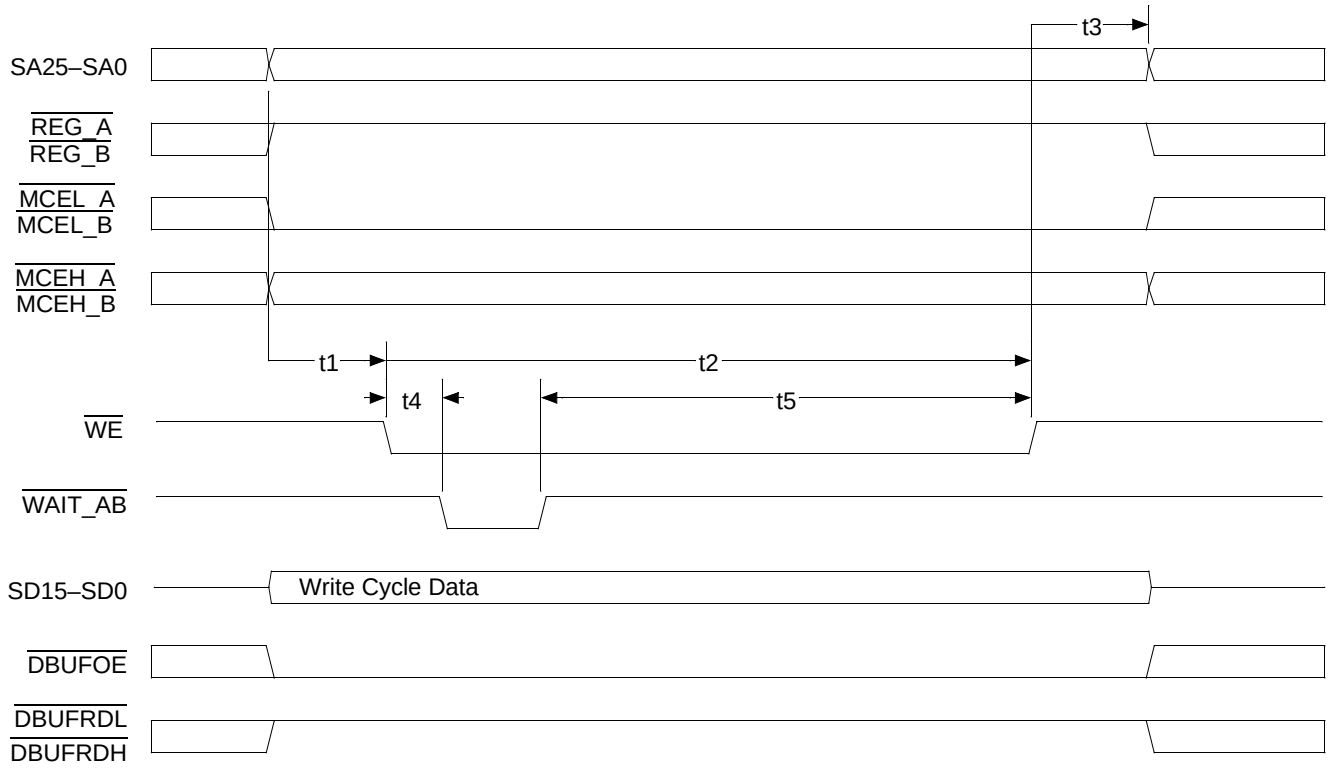


Figure 58. PC Card Common Memory Write Cycle (ÉlanSC400 Microcontroller Only)

Table 46. PC Card Common Memory Write Function (ÉlanSC400 Microcontroller Only)

Mode	$\overline{\text{REG\_x}}$	$\overline{\text{MCEH\_x}}$	$\overline{\text{MCEL\_x}}$	SA0	$\overline{\text{OE}}$	$\overline{\text{WE}}$	SD15–SD8	SD7–SD0
Byte Access	H H	H H	L L	L H	H H	L L	Indeterminate Indeterminate	Even byte Odd byte
Word Access	H	L	L	Indeterminate	H	L	Odd byte	Even byte
Odd-Byte-Only Access	H	L	H	Indeterminate	H	L	Odd byte	Indeterminate

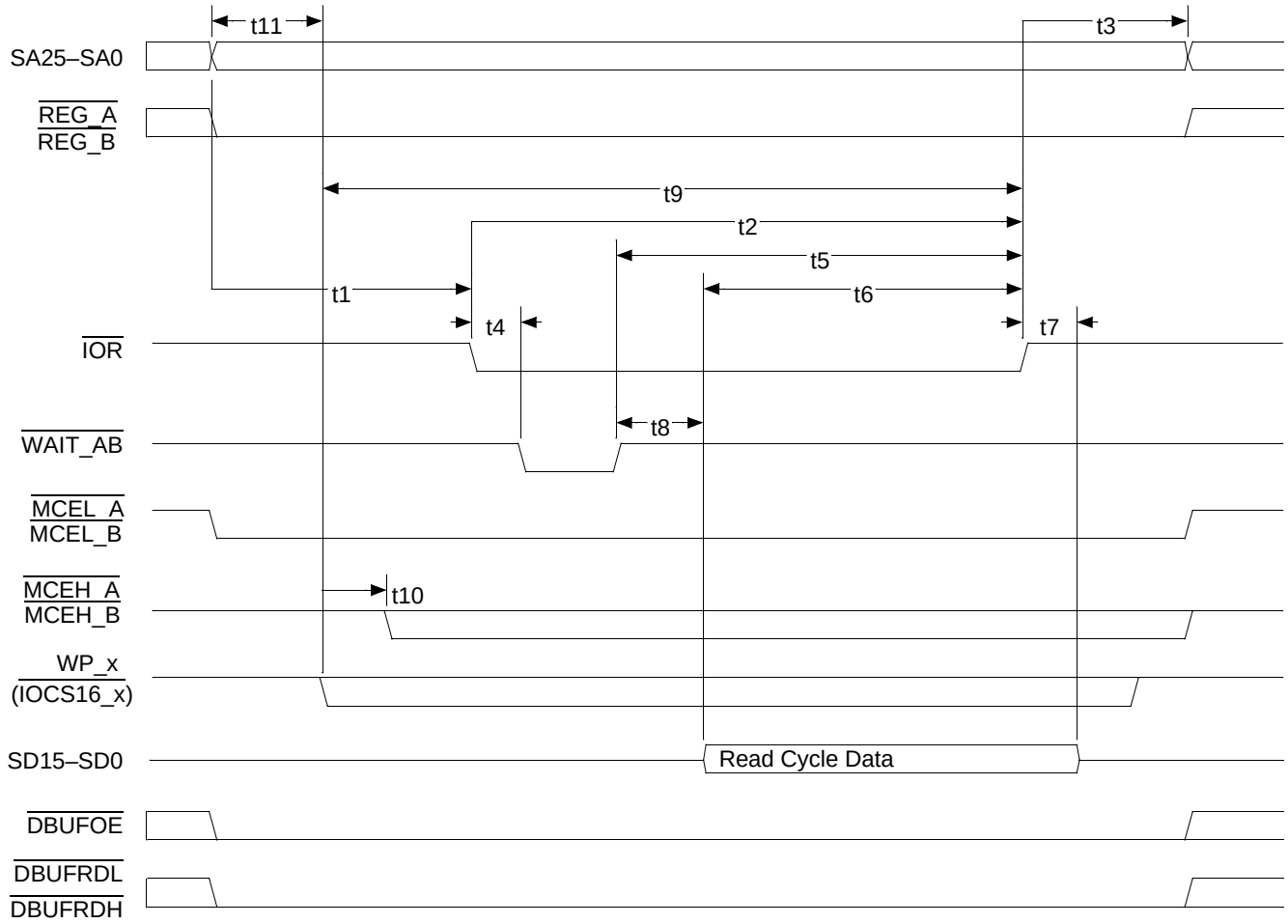


Figure 59. PC Card I/O Read Cycle

Table 47. PC Card I/O Read Function (ÉlanSC400 Microcontroller Only)

Mode	$\overline{\text{REG\_x}}$	$\overline{\text{MCEH\_x}}$	$\overline{\text{MCEL\_x}}$	SA0	$\overline{\text{IOR}}$	$\overline{\text{IOW}}$	SD15–SD8	SD7–SD0
Byte Access	L L	H H	L L	L H	L L	H H	Three-state Three-state	Even byte Odd byte
Word Access	L	L	L	Indeterminate	L	H	Odd byte	Even byte
High Byte Only	L	L	H	Indeterminate	L	H	Odd byte	Three-state

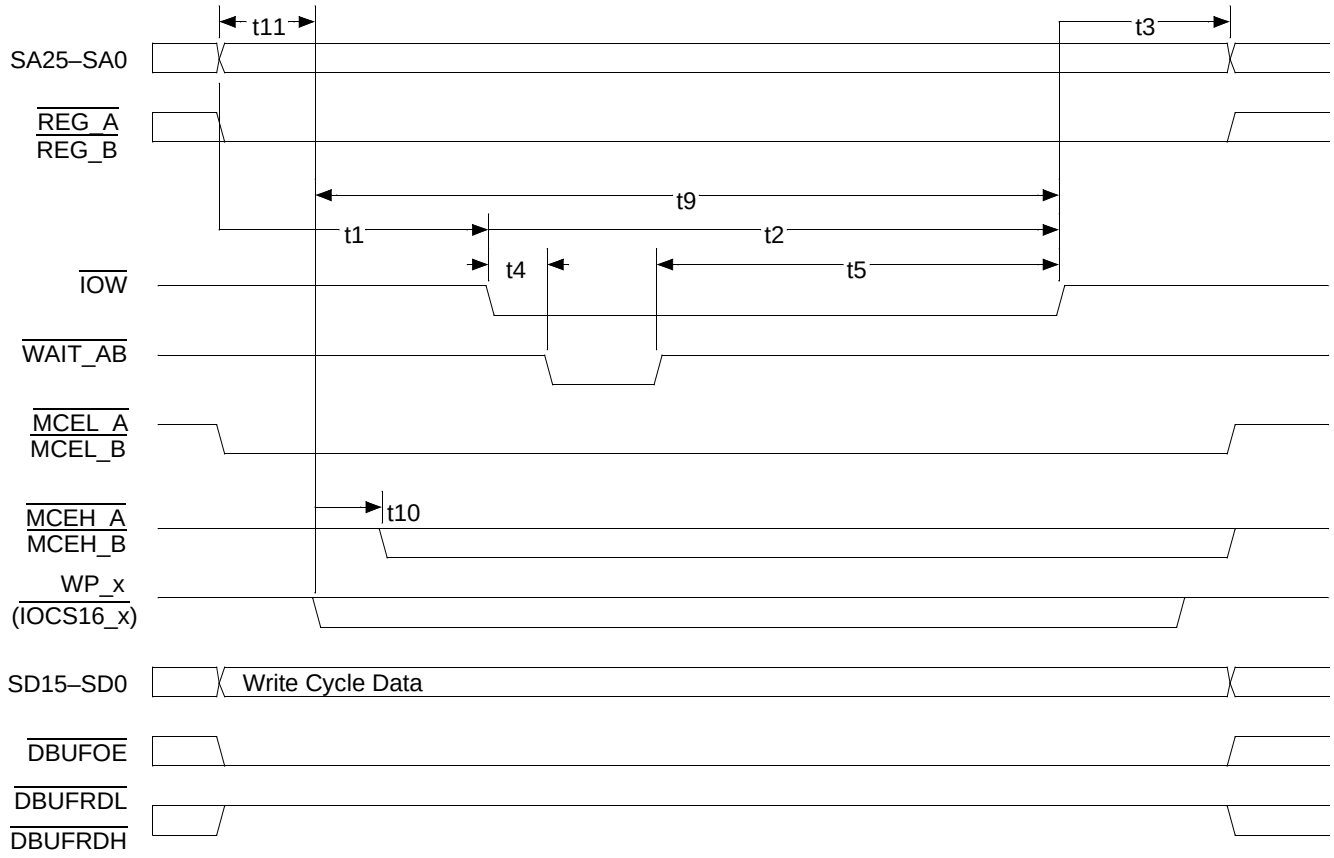


Figure 60. PC Card I/O Write Cycle

Table 48. PC Card I/O Write Function (ÉlanSC400 Microcontroller Only)

Mode	$\overline{\text{REG\_x}}$	$\overline{\text{MCEH\_x}}$	$\overline{\text{MCEL\_x}}$	SA0	$\overline{\text{IOR}}$	$\overline{\text{IOW}}$	SD15–SD8	SD7–SD0
Byte Access	L L	H H	L L	L H	H H	L L	Indeterminate Indeterminate	Even byte Odd byte
Word Access	L	L	L	Indeterminate	H	L	Odd byte	Even byte
Odd-Byte-Only Access	L	L	H	Indeterminate	H	L	Odd byte	Indeterminate

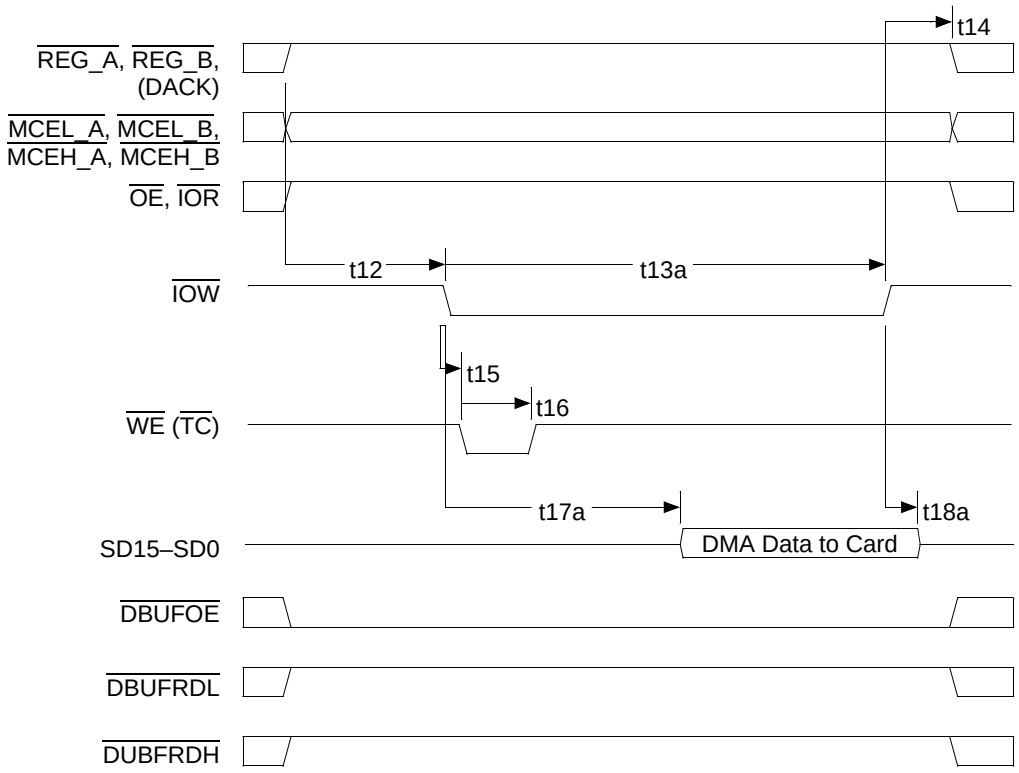


Figure 61. PC Card DMA Read Cycle (Memory Read to I/O Write)

Table 49. PC Card DMA Read Function (ÉlanSC400 Microcontroller Only)

Mode	DACK	DREQ	MCEH_x	MCEL_x	OE	WE	IOR	IOW	SD15–SD8	SD7–SD0
Byte Access	H	L	H	L	H	TC	H	L	Indeterminate	Even byte
Word Access	H	L	L	L	H	TC	H	L	Odd byte	Even byte

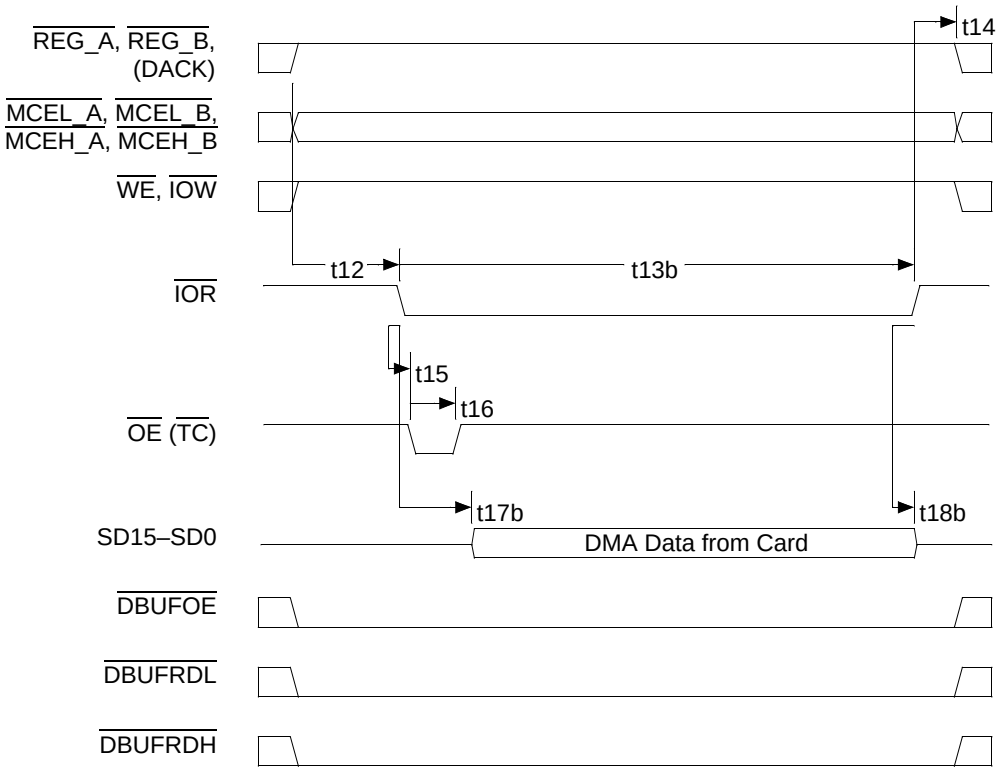


Figure 62. PC Card DMA Write Cycle (I/O Read to Memory Write)

Table 50. PC Card DMA Write Function (ÉlanSC400 Microcontroller Only)

Mode	DACK	DREQ	MCEH_x	MCEL_x	OE	WE	IOR	IOW	SD15–SD8	SD7–SD0
Byte Access	H	L	H	L	TC	H	L	H	Indeterminate	Even byte
Word Access	H	L	L	L	TC	H	L	H	Odd byte	Even byte

Table 51. LCD Graphics Controller Cycles—ÉlanSC400 Microcontroller Only

Symbol	Parameter Description	Notes	33-MHz External Bus		Unit
			Min	Max	
t1a	SCK period, monochrome panel	1	4T		ns
t1b	SCK period, color STN panel		2T		ns
t2	SCK High time	1	T		ns
t3	SCK Low time	1	T		ns
t4	Setup, data to SCK falling edge	1	T-15		ns
t5	Hold, LCD_data from SCK falling edge	1	T-15		ns
t6	Width, LC	1	8T		ns
t7	Setup, FRM to LC falling	1,2			ns
t8	Hold, FRM from LC falling	1,2			ns
t9	Delay, LC falling to M phase change		0	15	ns
t11a	Delay, power-on sequencing, $\overline{\text{LVDD}}$ to signals	3	7.8	62.5	ms
t11b	Delay, power-on sequencing, signals to $\overline{\text{LVEE}}$	4	7.8	62.5	ms
t12a	Delay, power-off sequencing, $\overline{\text{LVEE}}$ to signals, normal power-down	5	62.5	500	ms
t12b	Delay, power-off sequencing, signals to $\overline{\text{LVDD}}$, normal power-down	6	62.5	500	ms
t13	Delay, $\overline{\text{LVEE}}$ to LCD_SIGNALS off, emergency power-down		0		ns
t14	Delay, LCD_SIGNALS off to $\overline{\text{LVDD}}$ off, emergency power-down		0		ns
t15	Delay, emergency power-off sequencing from $\overline{\text{BL2}}$ edge		0	10	µs

Notes:

1. T = period of internal video dot clock—programmable via the Pixel Clock Control Register.

2. Programmable to within resolution of 8T intervals (single-screen mode) or 16T intervals (dual-screen mode).

3. Programmable through PMU Control Register 1, bits 2–0.

4. Programmable through PMU Control Register 1, bits 5–3.

5. Programmable through PMU Control Register 2, bits 2–0.

6. Programmable through PMU Control Register 2, bits 5–3.

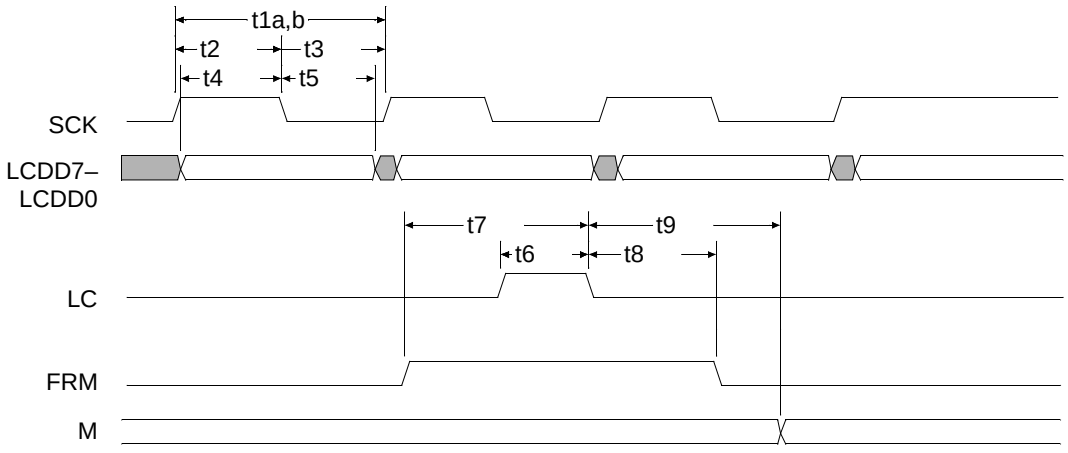


Figure 63. Graphics Panel Interface Timing (ÉlanSC400 Microcontroller Only)

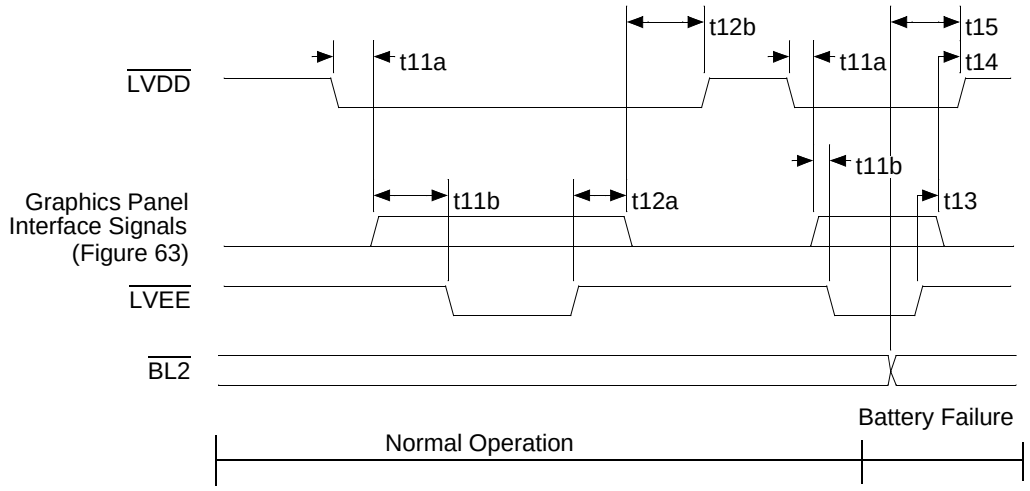


Figure 64. Graphics Panel Power Sequencing (ÉlanSC400 Microcontroller Only)

THERMAL CHARACTERISTICS

The thermal specifications for the ÉlanSC400 and ÉlanSC410 microcontrollers are given as a T_{CASE} (the case temperature) specification. The 33-MHz and 66-MHz devices are specified for operation when T_{CASE} is with the range of 0°C–+95°C. The 100-MHz device is specified for operation when T_{CASE} is within the range of 0°C–+85°C. T_{CASE} can be measured in any environment to determine whether the microcontroller is within specified operating range. The case temperature should be measured at the center of the top surface opposite the solder balls.

The ambient temperature (T_A) is guaranteed as long as T_{CASE} is not violated. The ambient temperature can be calculated from Ψ_{J-T} and θ_{JA} from the following equations:

$T_J = T_{CASE} + P \bullet \Psi_{J-T}$

$T_A = T_J - P \bullet \theta_{JA}$

$T_{CASE} = T_A + P \bullet (\theta_{JA} - \Psi_{J-T})$

where:

T_J is the junction temperature (°C).

T_A is the ambient temperature (°C).

T_{CASE} is the case temperature (°C).

Ψ_{J-T} is the junction-to-case thermal resistance (°C/W).

θ_{JA} is the junction-to-ambient thermal resistance (°C/W).

P is the maximum power consumption (W).

The values for θ_{JA} and Ψ_{J-T} are given in Table 52 for the BGA 292 package. These numbers are valid only for packages with all 292 balls soldered to a board with two power planes and two signal planes.

Table 53 shows the T_A allowable (without exceeding T_{CASE}) at various airflows and operating frequencies. P is calculated using the I_{CC} at 3.3 V as tabulated in the DC Characteristics section beginning on page 86.

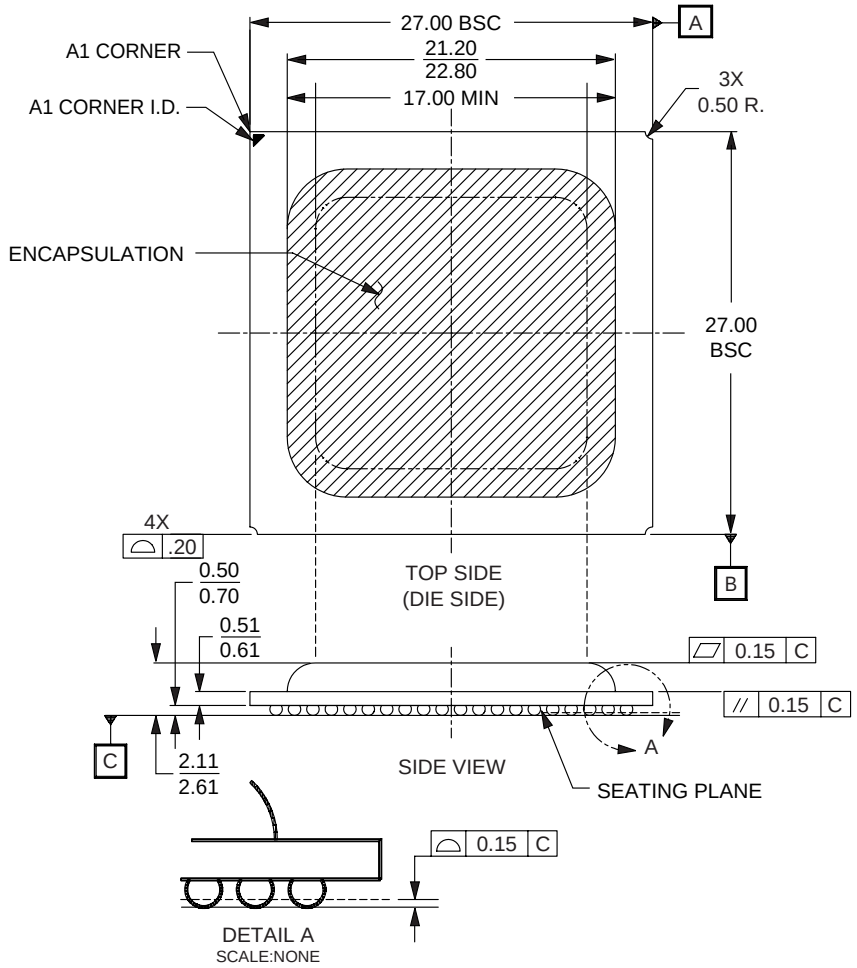
Table 52. Thermal Resistance Ψ_{J-T} and θ_{JA} (°C/W) for the 292-BGA Package)

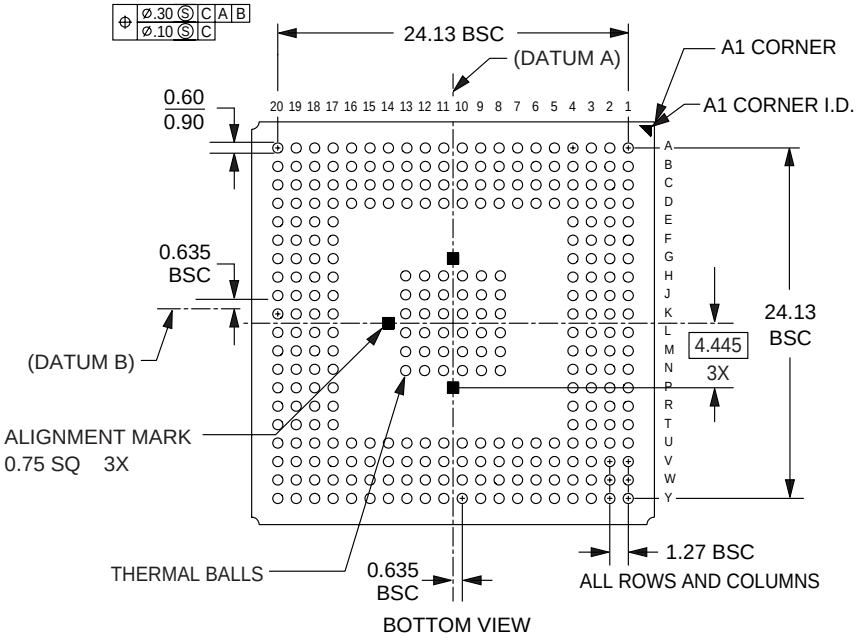
	Airflow in Feet/Minute (m/s)				
Thermal Resistance	**0 (0)**	**200 (1.01)**	**400 (2.03)**	**600 (3.04)**	**800 (4.06)**
θ_{JA}	25.0	20.5	19.0	18.1	17.4
Ψ_{J-T}	6.2	6.2	6.2	6.2	6.2

Table 53. Maximum T_A at Various Airflows in °C

	Airflow in Feet/Minute (m/s)				
Maximum T_A	**0 (0)**	**200 (1.01)**	**400 (2.03)**	**600 (3.04)**	**800 (4.06)**
at 33 MHz	83.7	86.4	87.3	87.8	88.3
at 66 MHz	72.0	77.5	79.3	80.4	81.3
at 100 MHz	50.8	59.0	61.7	63.4	64.6

PHYSICAL DIMENSIONS—BGA 292—PLASTIC BALL GRID ARRAY





16-038-BGA292-2_AB
ES114
9.14.98 lv

Trademarks

AMD, the AMD logo and combinations thereof, Am186, Am188, E86, K86, Élan, Comm86, and Systems in Silicon are trademarks, and Am386 and Am486 are registered trademarks of Advanced Micro Devices, Inc. FusionE86 is a service mark of Advanced Micro Devices, Inc. Microsoft and Windows are registered trademarks of Microsoft Corp. Other product names used in this publication are for identification purposes and may be trademarks of their respective companies.