AMENDMENT

Élan™SC400 Microcontroller

Register Set Reference Manual

Including Changes for the ÉlanSC410 Microcontroller

This document amends the *Élan™SC400 Microcontroller Register Set Reference Manual*, order #21032A, including additional information for the ÉlanSC410 microcontroller. It consists of three parts:

- "Documentation Defects and Corrections" lists documentation defects found in the original publication.
- "Changes for the ÉlanSC410 Microcontroller" on page 25 gives an overview of differences between the ÉlanSC400 microcontroller and the ÉlanSC410 microcontroller, and lists all the registers and bits not supported on the ÉlanSC410 microcontroller.
- The Index beginning on page 35 includes all of the registers listed in this document, with index offsets listed separately by subsystem.

DOCUMENTATION DEFECTS AND CORRECTIONS

Table 1 lists defects that have been found in the *Élan™SC400 Microcontroller Register Set Reference Manual,* order #21032A. Defects are listed in page order. Each entry lists the following:

- page number
- item to be corrected
- original text
- corrected text
- comments explaining the change

Table entries that correct text in a diagram do not contain the entire diagram. If a default value is changed, the original and changed default values are shown for the whole register. If other text is changed, the entry describes the affected part of the diagram and lists the original and changed text.

Each item description includes the register name, so changes affecting a particular register can be found by consulting the index in this document.

Unchanged portions of a paragraph are replaced by an ellipsis (...) in entries where this might make the change easier to find. The whole paragraph is included if it is useful for understanding why the change was made.

Table 1. Corrections to the *Élan™SC400 Microcontroller Register Set Reference Manual*

Page	Item	Original Text	Change To	Comment
Title Page		*Élan™SC400 Microcontroller Register Set Reference Manual*	*Élan™SC400 and ÉlanSC410 Microcontrollers Register Set Reference Manual*	Add ÉlanSC410 microcontroller to book title.
Chapter 2: PC/AT-Compatible Direct-Mapped Registers				
2-36	Programmable Interval Timer #1 Channel 0 Count Register, Address 0040h; last paragraph	When set up for either BCD or 16-bit binary count operation, the maximum count for Channel 0 is achieved by writing the internal counting element associated with this register to 0000h/d. See direct-mapped register 43h for more detail.	When set up for either BCD or 16-bit binary count operation, the maximum count for Channel 0 is achieved by writing the internal counting element associated with this register to 0. See direct-mapped register 43h for more detail.	Same change for all three pages. Notation "0000h/d" is misleading. Zero value written can be binary (hex) or BCD (decimal).
2-37	Programmable Interval Timer #1 Channel 1 Count Register, Address 0041h; last paragraph			
2-38	Programmable Interval Timer #1 Channel 2 Count Register, Address 0042h; last paragraph			

Publication# **21032** Rev: **A** Amendment/**1**
Issue Date: **December 1997**

Table 1. Corrections to the *Élan™SC400 Microcontroller Register Set Reference Manual* (Continued)

Page	Item	Original Text	Change To	Comment
2-49	PC/AT Keyboard Mouse Interface Status Register, Address 0064h; bit 5 description	When CSC index C0h[0] is set, this bit is a PC/2 mouse-compatible mouse output buffer full flay...	When CSC index C0h[0] is set, this bit is a PS/2 mouse-compatible mouse output buffer full flag...	Two corrections: “PS/2 mouse” and “flag.”
2-52	RTC/CMOS RAM Index Register, Address 0070h; diagram (bit 7 column)	Reserved – W	NMI_GATE 0 W	The NMI_GATE function has *not* moved to CSC index 9Dh[2] in the ÉlanSC400 or ÉlanSC410 microcontrollers.
	RTC/CMOS RAM Index Register, Address 0070h; bit 7 description	7 Reserved **Reserved** During read/modify/write operations, software must preserve this bit.	7 NMI_GATE **Master NMI Mask** 1 = NMI events are gated off from reaching the core 0 = NMI events will propagate to the CPU core	
	RTC/CMOS RAM Index Register, Address 0070h; Programming Notes	**Programming Notes** Bit 7 of this register is the master NMI gate control in a typical PC/AT Compatible system. For various reason, this bit has been made to reside at CSC index 9Dh[2]... (entire paragraph)	**Programming Notes**	
2-92	Master Software DRQ(n) Request Register, Address 00D2h; default bit values in diagram	0 0 0 0 0 0 0 0	x x x x x x x x	“x” = non-deterministic.
	Master Software DRQ(n) Request Register, Address 00D2h; bits 1–0 description	0 0 = Mask/unmask DMA Channel 4 mask per the REQDMA bit 0 1 = Mask/unmask DMA Channel 5 mask per the REQDMA bit 1 0 = Mask/unmask DMA Channel 6 mask per the REQDMA bit 1 1 = Mask/unmask DMA Channel 7 mask per the REQDMA bit	0 0 = Set/Reset DMA Channel 4 internal DMA request per the REQDMA bit 0 1 = Set/Reset DMA Channel 5 internal DMA request per the REQDMA bit 1 0 = Set/Reset DMA Channel 6 internal DMA request per the REQDMA bit 1 1 = Set/Reset DMA Channel 7 internal DMA request per the REQDMA bit	This field selects the DMA request channel to assert or deassert, depending on the state of bit 2.
2-93	Master DMA Mask Register Channels 4–7, Address 00D4h; default bit values in diagram	0 0 0 0 0 0 0 0	x x x x x x x x	“x” = non-deterministic.
2-94	Master DMA Mode Register Channels 4–7, Address 00D6h; default bit values in diagram	0 0 0 0 0 0 0 0	x x x x x x x x	“x” = non-deterministic.
2-95	Master DMA Clear Byte Pointer Register, Address 00D8h; default bit values in diagram	0 0 0 0 0 0 0 0	x x x x x x x x	“x” = non-deterministic.

Table 1. Corrections to the *Élan™SC400 Microcontroller Register Set Reference Manual* (Continued)

Page	Item	Original Text	Change To	Comment
2-96	Master DMA Controller Reset Register, Address 00DAh; default bit values in diagram	0 0 0 0 0 0 0 0	x x x x x x x x	"x" = non-deterministic.
2-97	Master DMA Controller Temporary Register, Address 00DAh; default bit values in diagram	0 0 0 0 0 0 0 0	x x x x x x x x	"x" = non-deterministic.
2-98	Master DMA Reset Mask Register, Address 00DCh; default bit values in diagram	0 0 0 0 0 0 0 0	x x x x x x x x	"x" = non-deterministic.
2-103 2-104	Parallel Port 2 Status Register, Address 0279h; default bit values in diagram	— — — — — — — —	? ? ? ? ? x x x	"x" = non-deterministic. "?" = depends on input state.
	Parallel Port 2 Status Register, Address 0279h; Programming Notes	**Programming Notes** (entire note)	**Programming Notes** On power-up, the parallel port defaults to PC/AT-Compatible mode. Upon switching to EPP mode the bit defaults change to ?????xx0b	Clarify behavior.
2-105	Parallel Port 2 Status Register (EPP Mode), Address 0279h; default bit values in diagram	— — — — — — — —	? ? ? ? ? x x 0	"x" = non-deterministic. "?" = depends on input state.
	Parallel Port 2 Status Register (EPP Mode), Address 0279h; Programming Notes	**Programming Notes**	**Programming Notes** On power-up, the parallel port defaults to PC/AT-Compatible mode. Upon switching to EPP mode the bit defaults change to ?????xx0b	Clarify behavior.

Table 1. Corrections to the *Élan™SC400 Microcontroller Register Set Reference Manual* (Continued)

Page	Item	Original Text	Change To	Comment
2-111	COM2 Baud Clock Divisor Latch LSB, Address 02F8h; default bit values in diagram	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1	Divisor must be nonzero.
	COM2 Baud Clock Divisor Latch LSB, Address 02F8h; Programming Notes	**Programming Notes**	**Programming Notes** In order to determine the UART baud rate that will result from a particular setting of the baud rate divisor latch, the 16x baud clock is divided by 16 to give a normalized baud rate. Because the standard 16x baud clock runs at 1.8432 MHz, the “normalized” baud rate is 115200. This rate is then divided by the value of the 16-bit baud clock divisor latch register. It is invalid to program a baud rate divisor of 0. The baud rate divisor latch, being a 16-bit value, provides for up to 64k–1 different baud rates (0 is not a valid baud rate). However, only a few of the available rates are ever used in common practice. See the following table for the commonly used baud rates along with the baud rate divisor required to achieve them: **Baud** / **Divisor Latch Value** 110 — 0417h 150 — 0300h 300 — 0180h 600 — 00C0h 1200 — 0060h 2400 — 0030h 4800 — 0018h 9600 — 000Ch 19200 — 0006h 38400 — 0003h 57600 — 0002h 115200 — 0001h	Add text to programming notes on both pages.
2-112	COM2 Baud Clock Divisor Latch MSB, Address 02F9h; Programming Notes			
2-114	COM2 Interrupt ID Register, Address 02FAh; bits 7–6 description	0 0 = No significance 0 1 = No significance 1 0 = 16450-compatible mode is enabled 1 1 = 16550-compatible mode is enabled	0 0 = 16450-Compatible mode is enabled (default) 0 1 = No significance 1 0 = No significance 1 1 = 16550-Compatible mode is enabled	Default of 0 0 indicates 16450-Compatible mode.
2-121	COM2 Modem Status Register, Address 02FEh; default bit values in diagram	x x x x 0 0 0 0	? ? ? ? 0 0 0 0	“?” = depends on input state

Table 1. Corrections to the *Élan™SC400 Microcontroller Register Set Reference Manual* (Continued)

Page	Item	Original Text	Change To	Comment
2-124 2-125	Parallel Port 1 Status Register, Address 0379h; default bit values in diagram	— — — — — — — —	? ? ? ? ? x x x	"x" = non-deterministic. "?" = depends on input state.
	Parallel Port 1 Status Register, Address 0379h; Programming Notes	**Programming Notes** (entire note)	**Programming Notes** On power-up, the parallel port defaults to PC/AT-Compatible mode. Upon switching to EPP mode the bit defaults change to ?????xx0b	Clarify behavior.
2-126	Parallel Port 1 Status Register (EPP Mode), Address 0379h; default bit values in diagram	— — — — — — — —	? ? ? ? ? x x 0	"x" = non-deterministic. "?" = depends on input state.
	Parallel Port 1 Status Register (EPP Mode), Address 0379h; Programming Notes	**Programming Notes**	**Programming Notes** On power-up, the parallel port defaults to PC/AT-Compatible mode. Upon switching to EPP mode the bit defaults change to ?????xx0b	Clarify behavior.

Table 1. Corrections to the *Élan™SC400 Microcontroller Register Set Reference Manual* (Continued)

Page	Item	Original Text	Change To	Comment
2-144	COM1 Baud Clock Divisor Latch LSB, Address 03F8h; default bit values in diagram	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1	Divisor must be nonzero
	COM1 Baud Clock Divisor Latch LSB, Address 03F8h; Programming Notes	**Programming Notes**	**Programming Notes** In order to determine the UART baud rate that will result from a particular setting of the baud rate divisor latch, the 16x baud clock is divided by 16 to give a normalized baud rate. Because the standard 16x baud clock runs at 1.8432 MHz, the “normalized” baud rate is 115200. This rate is then divided by the value of the 16-bit baud clock divisor latch register. It is invalid to program a baud rate divisor of 0. The baud rate divisor latch, being a 16-bit value, provides for up to 64k–1 different baud rates (0 is not a valid baud rate). However, only a few of the available rates are ever used in common practice. See the following table for the commonly used baud rates along with the baud rate divisor required to achieve them: **Baud** — **Divisor Latch Value** 110 — 0417h 150 — 0300h 300 — 0180h 600 — 00C0h 1200 — 0060h 2400 — 0030h 4800 — 0018h 9600 — 000Ch 19200 — 0006h 38400 — 0003h 57600 — 0002h 115200 — 0001h	Add text to programming notes on both pages.
2-145	COM1 Baud Clock Divisor Latch MSB, Address 03F9h; Programming Notes			
2-147	COM1 Interrupt ID Register, Address 03FAh; default value for bit 0 in diagram	0 0 = No significance 0 1 = No significance 1 0 = 16450-compatible mode is enabled 1 1 = 16550-compatible mode is enabled	0 0 = 16450-Compatible mode is enabled (default) 0 1 = No significance 1 0 = No significance 1 1 = 16550-Compatible mode is enabled	Default of 0 0 indicates 16450-Compatible mode.
2-150	COM1 Modem Control Register, Address 03FCh; bit 3 description	**Enable COM2 Interrupts**	**Enable COM1 Interrupts**	Correction.
2-153	COM1 Modem Status Register, Address 03FEh; default bit values in diagram	x x x x 0 0 0 0	? ? ? ? 0 0 0 0	“?” = depends on input state.

Table 1. Corrections to the *Élan™SC400 Microcontroller Register Set Reference Manual* (Continued)

Page	Item	Original Text	Change To	Comment
Chapter 3: Chip Setup and Control (CSC) Indexed Registers				
3-15	DRAM Control Register, Index 04h; Programming Notes	**Programming Notes**	**Programming Notes** The $\overline{CAS}$ precharge delay (controlled by TCP, bit 3) always has at least one added state (t) during CPU write-back and copy-back cycles. This is required due to the amount of time it takes for CPU data to be valid during subsequent cycles of a burst. Furthermore, if the DRAM width for the bank receiving the write-back or copy-back is 16 bits, the added wait state occurs *only* between DWORDs.	Add note text.
3-16	DRAM Refresh Control Register, Index 05h; bits 1–0 description	When the 32-KHz clock is used.... Thus, selecting the divide by 1 option results in a refresh interval of 15.6 ms. For the divide by 2, 4, and 8 selections, the refresh intervals are 31.2 ms, 62.5 ms, and 125 ms respectively.	When the 32-KHz clock is used... Thus, selecting the divide by 1 option results in a refresh interval of 15.6 µs. For the divide by 2, 4, and 8 selections, the refresh intervals are 31.2 µs, 62.5 µs, and 125 µs respectively.	Change all times to microseconds (µs).
3-20	Non-Cacheable Window 0 Address/ Attributes/SMM Register, Index 11h; default bit values in diagram	0 0 0 0 0 0 0 0	x 0 0 0 0 0 0 0	"x" = non-deterministic.
3-23	Cache and VL Miscellaneous Register, Index 14h; bit 4 VL_RESET description	**Vesa Local Bus Reset** 0 = $\overline{VL\_RESET}$ deasserted 1 = $\overline{VL\_RESET}$ asserted	**Vesa Local Bus Reset** 0 = $\overline{VL\_RST}$ signal deasserted 1 = $\overline{VL\_RST}$ signal asserted	Vesa local bus reset signal name is $\overline{VL\_RST}$.
3-26	Linear $\overline{ROMCS0}$/ Shadow Register, Index 21h; default bit values in diagram	x 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	Bit 7 default is zero. ***Note:*** *Bit 6 and bits 4–1 are reserved in the ÉlanSC410 microcontroller.*
3-31	$\overline{ROMCS0}$ Configuration Register B, Index 24h; bits 4–3 description	00 = 0 wait states	00 = Reserved—Not Valid	Zero wait states not supported for Fast ROM.
	$\overline{ROMCS0}$ Configuration Register B, Index 24h; bits 2–0 description	000 = 0 wait states	000 = Reserved—Not Valid	Zero wait states not supported for Fast ROM.
3-33	$\overline{ROMCS1}$ Configuration Register A, Index 25h; bits 2–1 description	These two bits can be read back to determine the $\overline{ROMCS1}$ data bus width which is set via pin strapping options, and latched at power-on reset.	These two bits can be read back to determine the $\overline{ROMCS1}$ data bus width.	Delete reference to pin strapping (which applies to $\overline{ROMCS0}$ only).

Table 1. Corrections to the *Élan™SC400 Microcontroller Register Set Reference Manual* (Continued)

Page	Item	Original Text	Change To	Comment
3-34	$\overline{\text{ROMCS1}}$ Configuration Register B, Index 26h; bits 4–3 description	00 = 0 wait states	00 = Reserved—Not Valid	Zero wait states not supported for Fast ROM.
	$\overline{\text{ROMCS1}}$ Configuration Register B, Index 26h; bits 2–0 description	000 = 0 wait states	000 = Reserved—Not Valid	Zero wait states not supported for Fast ROM.
3-36	$\overline{\text{ROMCS2}}$ Configuration Register A, Index 27h; bits 2–1 description	These two bits can be read back to determine the $\overline{\text{ROMCS2}}$ data bus width which is set via pin strapping options, and latched at power-on reset.	These two bits can be read back to determine the $\overline{\text{ROMCS2}}$ data bus width.	Delete reference to pin strapping (which applies to $\overline{\text{ROMCS0}}$ only).
3-37	$\overline{\text{ROMCS2}}$ Configuration Register B, Index 28h; bits 4–3 description	00 = 0 wait states	00 = Reserved—Not Valid	Zero wait states not supported for Fast ROM.
	$\overline{\text{ROMCS2}}$ Configuration Register B, Index 28h; bits 2–0 description	000 = 0 wait states	000 = Reserved—Not Valid	Zero wait states not supported for Fast ROM.
3-44	Pin Mux Register A, Index 38h; diagram, (bit 5 column)	Reserved 0 R/W	Reserved x	Correction.
	Pin Mux Register A, Index 38h; bit 5 description	**Reserved**	**Reserved** During read/modify/write operations, software must preserve this bit.	Correct omission.
	Pin Mux Register A, Index 38h; bits 2–1 descriptions	**Select GPIO... or ISA PIRQx** 0 = ... 1 = ISA signal is available...	**Select GPIO... or PIRQx** 0 = ... 1 = PIRQx signal...	"ISA PIRQ" is misleading. The PIRQ pins can be assigned to other interrupts.
	Pin Mux Register A, Index 38h; Programming Notes	**Programming Notes**	**Programming Notes** Bit 0 of this register must be set in order to use most ISA I/O peripherals. This is because it enables the AEN signal which, although listed in the DMA signal group, is generated by the DMA controller for consumption by all other ISA I/O devices which are not part of the current DMA transfer. These I/O devices should ignore I/O cycles on the bus if AEN is active because it means that fly-by DMA is occurring, not an I/O cycle.	Clarify. Bit 0 is not just for ISA DMA devices.

Table 1. Corrections to the *Élan™SC400 Microcontroller Register Set Reference Manual* (Continued)

Page	Item	Original Text	Change To	Comment
3-53	PMU Present and Last Mode Register, Index 41h; bit 6 description	Set this bit to immediately time out the current mode timer. This bit does not need to be cleared and is not read back. Software should not invoke this feature for at least 15 microseconds after a wake-up from Suspend mode. Software can read the last mode from bits 5–3 of this register and delay 15 microseconds if the last mode was Suspend mode before setting this bit.	Set this bit to force an immediate time-out of the current mode timer, causing the PMU to drop to the next lower speed. This bit does not need to be cleared and is not read back. This function is designed for debugging power management software. Software should not invoke this feature for at least one 32-KHz clock cycle (30.5 µs) after a wake-up from Suspend mode. Software can read the last mode from bits 5–3 of this register and delay 30.5 µs if the last mode was Suspend mode before setting this bit.	Expand description. Increase delay. Forced time-out might be missed (bit not latched) if used too soon after wake-up from Suspend mode.
3-57	Wake-Up Pause/High-Speed Clock Timers Register, Index 45h; bits 2–0 description	Timer value to count down after a wake up is sensed and the PLLs are started up (if necessary) and the GPIO_CSx signals are switched to High-Speed (or Low-Speed) mode levels (for those GPIO_CSx signals that are programmed to change based on PMU mode) to allow the power supplies to stabilize before the ÉlanSC400 microcontroller starts driving its outputs or using its inputs. Read returns the last value written.	Selects the required delay from the time a wake-up is sensed and the PLLs are started up to allow the power supplies to stabilize before the microcontroller begins driving its outputs or using its inputs. Read returns the last value written.	Rewrite. Remove reference to programmed GPIO_CSx signals. All chip functions are delayed.
3-59	Wake-Up Source Enable Register A, Index 52h; diagram, (bit 3 column)	Reserved 0 R/W	Reserved x	Correction.
	Wake-Up Source Enable Register A, Index 52h; bit 3 description	**Reserved**	**Reserved** During read/modify/write operations, software must preserve this bit.	Correct omission.
3-61	Wake-Up Source Enable Register C, Index 54h; diagram (bits 7–6 columns)	PDRQ1_WAKE PDRQ0_WAKE 0 0 R/W R/W	RESERVED RESERVED x x	DMA requests do not support wake-ups.
	Wake-Up Source Enable Register C, Index 54h; bit 7 description	PDRQ1_WAKE **Programmable DMA Request 1 (PDRQ1) Wake-Up Control** 0 = Do not wake up system 1 = Wake up system	Reserved **Reserved**	DMA requests do not support wake-ups.
	Wake-Up Source Enable Register C, Index 54h; bit 6 description	PDRQ0_WAKE **Programmable DMA Request 0 (PDRQ0) Wake-Up Control** 0 = Do not wake up system 1 = Wake up system	Reserved **Reserved**	DMA requests do not support wake-ups.

Table 1. Corrections to the *Élan™SC400 Microcontroller Register Set Reference Manual* (Continued)

Page	Item	Original Text	Change To	Comment
3-63	Wake-Up Source Status Register A, Index 56h; diagram, (bit 3 column)	Reserved 0 R/W	Reserved x	Correction.
	Wake-Up Source Status Register A, Index 56h; bit 3 description	**Reserved**	**Reserved** During read/modify/write operations, software must preserve this bit.	Correct omission.
3-65	Wake-Up Source Status Register C, Index 58h; diagram (bits 7–6 columns)	PDRQ1_WOKE PDRQ0_WOKE 0 0 R/W R/W	RESERVED RESERVED x x	DMA requests do not support wake-ups.
	Wake-Up Source Status Register C, Index 58h; bit 7 description	PDRQ1_WOKE **Programmable DMA Request 1 (PDRQ1) Wake-Up Status** Write to ‘0b to clear. 0 = Did not wake up system 1 = Awakened system	Reserved **Reserved**	DMA requests do not support wake-ups.
	Wake-Up Source Status Register C, Index 58h; bit 6 description	PDRQ0_WOKE **Programmable DMA Request 0 (PDRQ0) Wake-Up Status** Write to ‘0b’ to clear. 0 = Did not wake up system 1 = Awakened system	Reserved **Reserved**	DMA requests do not support wake-ups.
3-73	Activity Source Enable Register C, Index 64h; diagram (bit 6 column)	DRQ_IS_ACT 0 R/W	RESERVED x	DMA requests do not support activities.
	Activity Source Enable Register C, Index 64h; space below diagram	\		Remove stray backslash character.
	Activity Source Enable Register C, Index 64h; bit 6 description	DRQ_IS_ACT **DMA Request Will Be Activity** 0 = Not an activity 1 = Will be activity	Reserved **Reserved**	DMA requests do not support activities.
3-74	Activity Source Enable Register D, Index 65h; diagram (bit 7)	INTREG_WAS_ACT	INTREG_IS_ACT	Correction.
	Activity Source Enable Register D, Index 65h; bit 7 description	INTREG_WAS_ACT	INTREG_IS_ACT	Correction.

Table 1. Corrections to the *Élan™SC400 Microcontroller Register Set Reference Manual* (Continued)

Page	Item	Original Text	Change To	Comment
3-77	Activity Source Status Register C, Index 68h; diagram (bit 6 column)	DRQ_WAS_ACT 0 R/W	RESERVED x	DMA requests do not support activities.
	Activity Source Status Register C, Index 68h; bit 6 description	DRQ_WAS_ACT **DMA Request Activity Status** Write to '0b' to clear. 0 = Not detected as activity 1 = Detected as activity	Reserved **Reserved**	DMA requests do not support activities.
3-81	Activity Classification Register C, Index 6Ch; diagram (bit 6 column)	DRQ_SEC_ACT 0 R/W	RESERVED x	DMA requests do not support activities.
	Activity Classification Register C, Index 6Ch; bit 6 description	DRQ_SEC_ACT **DMA Request Activity Status** 0 = Primary activity 1 = Secondary activity	Reserved **Reserved**	DMA requests do not support activities.
3-85	Battery/AC Pin Configuration Register B, Index 71h; bit 0 description, last paragraph	...to enter Suspend mode.8	...to enter Suspend mode.	Remove stray character.
3-86	Battery/AC Pin State Register, Index 72h; bits 4–0 default values in diagram	x x x 0 0 0 0 0	x x x ? ? ? ? ?	"x" = non-deterministic. "?" = depends on input state.
3-87	CPU Clock Speed Register, Index 80h; bit 0 description	1 = Clock tripled (99 MHz) The ÉlanSC400 microcontroller and ÉlanSC410 microcontroller may require special packaging to safely support clock-tripled mode. Selection of clock-tripled mode results in much higher heat generation by the device. Selecting clock-tripled mode for an ÉlanSC400 microcontroller device which uses a package that is not specifically approved by AMD for use at clock-tripled speed may result in erratic system operation, loss of data, or damage to the ÉlanSC400 microcontroller device.	1 = Clock tripled (100 MHz) Selection of clock-tripled mode results in much higher heat generation by the device. ÉlanSC400 and ÉlanSC410 microcontroller devices that use the 292-pin BGA (Ball Grid Array) package type can safely support clock-tripled mode under AMD-approved operating conditions. AMD may make the ÉlanSC400 and ÉlanSC410 microcontroller devices available in other package types that are not approved by AMD for use at clock-tripled speed. Selecting clock-tripled mode for such devices may result in erratic system operation, loss of data, or damage to the microcontroller device.	Correction. Rewrite.

Table 1. Corrections to the *Élan™SC400 Microcontroller Register Set Reference Manual* (Continued)

Page	Item	Original Text	Change To	Comment
3-88	CPU Clock Auto Slowdown Register, Index 81h; bit 4 description	The PMU modes or states provide a way to trade off performance for processing power in discrete steps. The auto slowdown feature allows the (average power/average performance) to be fine tuned when the PMU is operating in the highest available PMU mode. This can be either be Hyper-Speed mode or High-Speed mode, depending on whether or not Hyper-Speed mode is disabled. Enabling this feature causes the CPU clock to switch between the fast speed and slow speed operating frequencies as defined by the FAST_PERIOD and SLOW_PERIOD bit fields.	The PMU modes or states provide a way to trade off performance for processing power in discrete steps. The auto slowdown feature allows the (average power/average performance) to be fine tuned by allowing the PMU to toggle between High-Speed and Low-Speed PMU modes with a duty cycle which is user configurable via the FAST_PERIOD and SLOW_PERIOD bit fields in this register. The auto slowdown feature must not be enabled if Hyper-Speed mode is enabled, or unexpected system operation may occur.	Auto slowdown is not supported in Hyper-Speed mode.
	CPU Clock Auto Slowdown Register, Index 81h; bits 3–2 description	When Auto Slowdown is enabled, these bits define how long the CPU clock runs at the slow clock frequency before switching up to run at the fast clock frequency. When the Hyper-Speed PMU mode is enabled, the slow clock frequency will be the High-Speed mode clock frequency. When the Hyper-Speed PMU mode is disabled, the slow clock will equal the Low-Speed mode clock frequency. See the FAST_PERIOD field for further explanation.	When Auto Slowdown is enabled, these bits define how long the CPU clock runs at the clock frequency that is currently configured for Low-Speed PMU mode before switching up to run at the clock frequency that is currently configured for High-Speed PMU mode. See the FAST_PERIOD field for further explanation.	Auto slowdown is not supported in Hyper-Speed mode.
	CPU Clock Auto Slowdown Register, Index 81h; bits 1–0 description	When Auto Slowdown is enabled, these bits define how long the CPU clock runs at the fast clock frequency before switching down to run at the slow clock frequency. When the Hyper-Speed PMU mode is enabled, the fast clock frequency will be the Hyper-Speed mode clock frequency. When the Hyper-Speed PMU mode is disabled, the fast clock will equal the High-Speed mode clock frequency. See the SLOW_PERIOD field for further explanation.	When Auto Slowdown is enabled, these bits define how long the CPU clock runs at the clock frequency that is currently configured for High-Speed PMU mode before switching down to run at the clock frequency that is currently configured for Low-Speed PMU mode. See the SLOW_PERIOD field for further explanation.	Auto slowdown is not supported in Hyper-Speed mode.
3-90	Clock Control Register, Index 82h; default bit values in diagram	0 0 1 0 0 0 0 1	x 0 1 0 0 0 0 1	"x" = non-deterministic.
	Clock Control Register, Index 82h; diagram (bit 7 R/W status)	R/W		Bit 7 R/W status is undefined.
3-91	CLK_IO Pin Output Clock Select Register, Index 83h; bits 3–0 description	**CLI_IO Pin Select** ... 0 0 0 0 = UART clock (18.432 MHz)	**CLK_IO Pin Select** ... 0 0 0 0 = UART clock (1.8431 MHz)	Correct pin name and UART clock frequency.
3-94	Miscellaneous SMI/ NMI Enable Register, Index 90h; bit 2 description	**Rising Edge on SUS_RES Pin XMI Enable**	**Falling Edge on SUS_RES Pin XMI Enable**	Correction.

Table 1. Corrections to the *Élan™SC400 Microcontroller Register Set Reference Manual* (Continued)

Page	Item	Original Text	Change To	Comment
3-94	Miscellaneous SMI/ NMI Enable Register, Index 90h; Programming Notes	**Programming Notes** **Bits 6–2:** If, when an XMI is enabled... to XMI on a falling edge, the SMI will occur immediately... 1. Disable the SMI at its master control. ... 4. Re-enable the SMI at its master source...	**Programming Notes** **Bits 6–2:** If, when an XMI is enabled... to XMI on a falling edge, the XMI will occur immediately... 1. Disable the XMI at its master control. ... 4. Re-enable the XMI at its master control...	Replace “SMI” with “XMI” (three places) and replace “master source” with “master control” (one place).
3-97	Battery Low and ACIN SMI/NMI Enable Register, Index 93h; diagram and description (all bits)	ACN_RISE_WAS_XMI ACN_FALL_WAS_XMI BL2_RISE_WAS_XMI BL2_FALL_WAS_XMI BL1_RISE_WAS_XMI BL1_FALL_WAS_XMI BL0_RISE_WAS_XMI BL0_FALL_WAS_XMI	ACN_RISE_WILL_XMI ACN_FALL_WILL_XMI BL2_RISE_WILL_XMI BL2_FALL_WILL_XMI BL1_RISE_WILL_XMI BL1_FALL_WILL_XMI BL0_RISE_WILL_XMI BL0_FALL_WILL_XMI	Change “was” to “will” in all bit names. This is an enable register, not a status register.
	Battery Low and ACIN SMI/NMI Enable Register, Index 93h; description (all bits)	0 = Did not cause SMI/NMI 1 = Caused SMI/NMI	0 = Do not cause SMI/NMI 1 = Cause SMI/NMI	Change state definitions for all bits. This is an enable register, not a status register.
3-103	Battery Low and ACIN SMI/NMI Status Register, Index 97h; Programming Notes, step 4 (last paragraph)	Re-enable the XMI at its master control (CSC index 9Dh[2] for NMIs...	Re-enable the XMI at its master control (port 70h[7] for NMIs...	The NMI_GATE function has *not* moved to CSC index 9Dh[2] in the ÉlanSC400 or ÉlanSC410 microcontrollers.
3-104	SMI/NMI Select Register, Index 98h; bit 3 description	...Each of these can have individual enables. See the keyboard index registers for more detail.	...Each of these can be individually enabled. See the PC Card and Keyboard SMI/NMI Enable Register, CSC Index 91h, for more detail.	Clarify.

Table 1. Corrections to the *Élan™SC400 Microcontroller Register Set Reference Manual* (Continued)

Page	Item	Original Text	Change To	Comment
3-105	I/O Access SMI Enable Register A, Index 99h; descriptions for bits 4–0	1 = Cause an SMI due to....	1 = Cause an SMI due to.... Requires special handling; see Programming Notes section.	For all five bits, add reference to programming notes.
	I/O Access SMI Enable Register A, Index 99h; Programming Notes	**Programming Notes**	**Programming Notes** Special handling for SMI to emulate I/O accesses: If the system requires the use of emulating and/or restarting I/O accesses, then the software must select the 80486 cache policy as write through via CSC index 14h, bit 0. *Do not* use the I/O Trap address in SMBASE offset 0FF06h to determine which I/O address to emulate. Instead, to determine the I/O address to emulate, the software should examine the saved values of the EIP and CS at SMBASE offsets 0FFF0h and 0FFACh, respectively. The software then examines the instruction that caused the SMI by decrementing the EIP value, determining the opcode, and using the DX and AX registers in the SMM save state or opcode operands to determine the correct I/O address. The I/O restart (writing 0FFh to SMBASE offset 0FF00h) will work correctly with the write through policy.	The ÉlanSC400 microcontroller only supports I/O trapping with the write through cache policy and requires special handling.

Table 1. Corrections to the *Élan™SC400 Microcontroller Register Set Reference Manual* (Continued)

Page	Item	Original Text	Change To	Comment
3-106	I/O Access SMI Enable Register B, Index 9Ah; descriptions for bits 6–0	1 = Cause an SMI due to....	1 = Cause an SMI due to.... Requires special handling; see Programming Notes section.	For all seven bits, add reference to programming notes.
	I/O Access SMI Enable Register B, Index 9Ah; Programming Notes	**Programming Notes**	**Programming Notes** Special handling for SMI to emulate I/O accesses: If the system requires the use of emulating and/or restarting I/O accesses, then the software must select the 80486 cache policy as write through via CSC index 14h, bit 0. *Do not* use the I/O Trap address in SMBASE offset 0FF06h to determine which I/O address to emulate. Instead, to determine the I/O address to emulate, the software should examine the saved values of the EIP and CS at SMBASE offsets 0FFF0h and 0FFACh, respectively. The software then examines the instruction that caused the SMI by decrementing the EIP value, determining the opcode, and using the DX and AX registers in the SMM save state or opcode operands to determine the correct I/O address. The I/O restart (writing 0FFh to SMBASE offset 0FF00h) will work correctly with the write through policy.	The ÉlanSC400 microcontroller only supports I/O trapping with the write through cache policy and requires special handling.
3-109	XMI Control Register, Index 9Dh; diagram, bits 7–3 and bit 2 functions	Reserved NMI_GATE	Reserved	Add bit 2 to Reserved field bits 7–3. Delete bit 2 description. The NMI_GATE function is performed by direct-mapped Port 70h[7] in the ÉlanSC400 and ÉlanSC410 microcontrollers.
	XMI Control Register, Index 9Dh; bits 7–3 description	7–3 Reserved...	7–2 Reserved...	
	XMI Control Register, Index 9Dh; bit 2 description	2 NMI_GATE **Master NMI Enable**... (entire description)		
3-116	GPIO Read-Back/Write Register A–D, Index A6h, A7h, A8h, A9h; default bit values in diagram	x x x x x x x x	? ? ? ? ? ? ? ?	"?" = depends on input state.
3-122	GPIO_PMUB Mode Change Register, Index ABh; bit 4 description	**Drive GPIO_PMUB Signal with Programmed Value in Hyper-Speed Mode**	**Drive GPIO_PMUB Signal with Programmed Value in High-Speed Mode**	Correction.

Table 1. Corrections to the *Élan™SC400 Microcontroller Register Set Reference Manual* (Continued)

Page	Item	Original Text	Change To	Comment
3-124	GPIO_PMUC Mode Change Register, Index ACh; bit 4 description	**Drive GPIO_PMUC Signal with Programmed Value in Hyper-Speed Mode**	**Drive GPIO_PMUC Signal with Programmed Value in High-Speed Mode**	Correction.
3-126	GPIO_PMUD Mode Change Register, Index ADh; bit 4 description	**Drive GPIO_PMUD Signal with Programmed Value in Hyper-Speed Mode**	**Drive GPIO_PMUD Signal with Programmed Value in High-Speed Mode**	Correction.
3-130	GPIO_XMI to GPIO_CS Map Register, Index B0h; bits 3–0 description, step 1 of procedure	1. Disable XMIs to the CPU core... For NMIs, this is done via CSC index 9Dh[2].	1. Disable XMIs to the CPU core... For NMIs, this is done via port 70h[7].	The NMI_GATE function has *not* moved to CSC index 9Dh[2] in the ÉlanSC400 or ÉlanSC410 microcontrollers.
3-135	GP_CSA I/O Address Decode and Mask Register, Index B5h; diagram (bits 1–0 R/W status)		R/W	Field is R/W.
3-137	GP_CSB I/O Address Decode and Mask Register, Index B7h; diagram (bits 1–0)	CSB_SA0_MASK	CSB_ADDR[9–8]	Correction. Bit 2 name was repeated in bits 1–0 field.
3-147	Keyboard Configuration Register A, Index C0h; bit 2 description	... 1 = Disabled	... 1 = Disabled This bit may not be set if XT mode is enabled via CSC register C1h[4].	Addition.
3-150	Keyboard Configuration Register B, Index C1h; bit 1 description, fourth line	... transmit date, this bit...	... transmit data, this bit...	Correction.
3-164	Internal I/O Device Disable/Echo Z-Bus Configuration Register, Index D0h; bit 5 description	... During ECHO_ZBUS cycles, AEN is also asserted so that ISA I/O devices will not decode the ECHO_ZBUS cycles.	... During echoed cycles, AEN is also asserted so that ISA I/O devices will not decode the echoed cycles.	Use more meaningful references to internal I/O cycles.
3-165	Internal I/O Device Disable/Echo Z-Bus Configuration Register, Index D0h; bit 2 description	**DMA 0 Slave Controller Disable**	**DMA 0 Master Controller Disable**	Correction.
	Internal I/O Device Disable/Echo Z-Bus Configuration Register, Index D0h; bit 1 description	1 = Internal PC Card controller... ...If the last value written to port 3E0h was greater than 80h, the write to 3E1h will generate an external bus cycle.	1 = Internal PC Card controller... ...If the last value written to port 3E0h was equal to or greater than 80h, the write to 3E1h will generate an external bus cycle.	Correction.

Table 1. Corrections to the *Élan™SC400 Microcontroller Register Set Reference Manual* (Continued)

Page	Item	Original Text	Change To	Comment
3-168	Parallel Port Configuration Register, Index D2h; bits 1–0 description	0 0 = Set/reset DMA Channel 4 DMA request per the REQDMA bit 0 1 = Set/reset DMA Channel 5 DMA request per the REQDMA bit 1 0 = Set/reset DMA Channel 6 DMA request per the REQDMA bit 11= Set/reset DMA Channel 7 DMA request per the REQDMA bit	0 0 = Unidirectional mode (PC/AT compatible) 0 1 = EPP mode 1 0 = Bidirectional mode (PS/2 compatible) 11= Reserved	Correction.
3-169	UART FIFO Control Shadow Register, Index D3h; diagram (bits 7–6 R/W status)		R	Bits 7–6 R/W status is R.
	UART FIFO Control Shadow Register, Index D3h; diagram (bits 5–4 and bit 3 function)	Reserved Reserved	Reserved	Combine fields for bits 5–3.
	UART FIFO Control Shadow Register, Index D3h; diagram (bit 3 R/W status)	R		Bit 3 R/W status is undefined.
	UART FIFO Control Shadow Register, Index D3h; bits 5–4 and bit 3 descriptions	(entire description text)	5–3 Reserved **Reserved** During read/modify/write operations, software must preserve these bits.	Combine descriptions for bits 5–3 and correct.
3-174	Interrupt Configuration Register E, Index D8h; diagram, (bits 2–0 column)	Reserved 0 0 0 R/W	Reserved x x x	Correction.
	Interrupt Configuration Register E, Index D8h; bits 2–0 description	**Reserved**	**Reserved** During read/modify/write operations, software must preserve this bit.	Correct omission.
3-184	Suspend Pin State Register A, Index E3h; diagram, (bit 4 column)	Reserved 0 R/W	Reserved x	Correction.
	Suspend Pin State Register A, Index E3h; bit 4 description	**Reserved**	**Reserved** During read/modify/write operations, software must preserve this bit.	Correct omission.

Table 1. Corrections to the *Élan™SC400 Microcontroller Register Set Reference Manual* (Continued)

Page	Item	Original Text	Change To	Comment
3-189	IrDA Control Register, Index EAh; bit 0 description	0 = UART mode 1 = IrDA mode When UART mode is selected, the IrDA interface and all associated control and status bits have no meaning except this bit. When IrDA mode is selected and is operated in Slow-Speed IrDA mode by clearing this bit, use all of the normal 16550 UART control and status bits to transmit and receive data over the IrDA interface.	0 = UART mode. When UART mode is selected, the IrDA interface and all associated control and status bits have no meaning except this bit. Serial data transfer is via the SIN and SOUT pins. 1 = IrDA mode. When IrDA mode is selected, the IrDA interface is enabled, and data transfer will occur over the dedicated SIROUT and SIRIN pins. Note: Software can switch between UART and IrDA modes at any time.	Rewrite.
	IrDA Control Register, Index EAh; Programming Notes	The use of either Low- or High-Speed IrDA requires that the on-board UART be enabled by setting CSC index D1h[0].	The use of either Slow- or High-Speed IrDA requires that the on-board UART be enabled by setting CSC index D1h[0]. If CSC index EAh[3] = ‘0b,’ the interrupt status bits in this register are undefined. Thus, they cannot be used for polled status unless the High-Speed IrDA interrupt is enabled. To use these bits in a polled-only fashion, simply set CSC index D8h[6–5] = ‘00b’ while CSC index EAh[3] = ‘1b.’	Rewrite.
3-192	IrDA CRC Status Register, Index ECh; default bit values in diagram	x 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	Bit 7 default is zero.

Table 1. Corrections to the *Élan™SC400 Microcontroller Register Set Reference Manual* (Continued)

Page	Item	Original Text	Change To	Comment
3-194	IrDA Frame Length Register A, Index EEh; bits 7–0 description	Bits 7–0 of the 11-bit IrDA Data Frame Length Register. See CSC index EFh for bits 10–8. The 11-bit value... (entire description)	Bits 7–0 of the 12-bit IrDA Data Frame Length Register. See CSC index EFh for bits 11–8. The 12-bit value written across these two registers controls the length of a transmitted IrDA data frame. This 12-bit internal register should be programmed with the total number of frame bytes (address, control, and information fields) plus two for the 2-byte CRC/FCS field. The programmed value must not include the two required BOF bytes or the STO flag byte. For example, consider a frame to be transmitted that contains the following bytes, which must be written into the frame's (DRAM-based) DMA transmit buffer prior to starting the transmit process. Assuming that the DMA memory address will increment after each transfer (see the direct-mapped DMA Mode Registers), the frame will appear in the buffer as follows: (lower addresses) BOF/STA = 2 bytes ADDR = 1 byte Control = 1 byte Information = 19 bytes CRC/FCS = 2 bytes EOF/STO = 1 byte (higher addresses) Total number of bytes in the transmit buffer = 26 Subtracting three (for the two required BOF bytes plus the one required STO byte) from the total results in 23d (17h). Thus, the value of 17h should be programmed into the 12-bit Frame Length Register via CSC Index EEh–EFh.	Modify for frame length of 12 bits rather than 11. Expand description.

Table 1. Corrections to the *Élan™SC400 Microcontroller Register Set Reference Manual* (Continued)

Page	Item	Original Text	Change To	Comment
3-195	IrDA Frame Length Register B, Index EFh; bit names in diagram	Bits 7–3: Reserved Bits 2–0: FRAME_LEN[10–8]	Bits 7–4: Reserved Bits 3–0: FRAME_LEN[11–8]	Modify for frame length of 12 bits rather than 11.
	IrDA Frame Length Register B, Index EFh; bit default values in diagram	x x x x x 0 0 0	x x x x 0 0 0 0	
	IrDA Frame Length Register B, Index EFh; bit name text	Bits 7–3: Reserved Bits 2–0: FRAME_LEN[10–8]	Bits 7–4: Reserved Bits 3–0: FRAME_LEN[11–8]	
	IrDA Frame Length Register B, Index EFh; bit descriptions	**IrDA Data Frame Length... Bits 10–8** Bits 10–8 of the 11-bit IrDA Data Frame Length Register. See CSC index EFh for bits 7–0. The 11-bit value... the number programmed in this 11-bit register will be data size plus 3... the length of each of these frames must conform to FRAME_LEN10–FRAME_LEN0.	**IrDA Data Frame Length... Bits 11–8** Bits 11–8 of the 12-bit IrDA Data Frame Length Register. See CSC index EEh for bits 7–0. The 12-bit value... the number programmed in this 12-bit register will be data size plus three... the length of each of these frames must conform to FRAME_LEN11–FRAME_LEN0.	Modify for frame length of 12 bits rather than 11. Also index value referred to should be EEh, not EFh.
3-201	ÉlanSC400 Microcontroller Revision ID Register, Index FFh; default bit values in diagram	0 0 0 0 0 0 0 0	— — — — — — — —	Default values depend on the microcontroller version.
	ÉlanSC400 Microcontroller Revision ID Register, Index FFh; Programming notes	**Programming Notes** The value read back depends on the current major.minor versions of a specific ÉlanSC400 microcontroller.	**Programming Notes** The value read back depends on the current major.minor versions of a specific ÉlanSC400 microcontroller. Contact your AMD representative for current version information.	Addition.

Table 1. Corrections to the *Élan™SC400 Microcontroller Register Set Reference Manual* (Continued)

Page	Item	Original Text	Change To	Comment
Chapter 4: RTC and CMOS RAM Indexed Registers				
4-3	RTC/CMOS RAM Index Register, Address 0070h; diagram (bit 7 column)	Reserved – 	NMI_GATE 0 W	The NMI_GATE function has *not* moved to CSC index 9Dh[2] in the ÉlanSC400 or ÉlanSC410 microcontrollers.
	RTC/CMOS RAM Index Register, Address 0070h; bit 7 description	7 Reserved **Reserved**	7 NMI_GATE **Master NMI Enable** 1 = NMI events are gated off from reaching the core 0 = NMI events will propagate to the CPU core	
	RTC/CMOS RAM Index Register, Address 0070h; Programming Notes	**Programming Notes** Bit 7 of this register is the master NMI gate control in a typical PC/AT Compatible system. For various reason, this bit has been made to reside at CSC index 9Dh[2]... (entire paragraph)	**Programming Notes**	
4-15	General Purpose CMOS RAM, Indexes 0E–7Fh	—	—	Move section to end of chapter.
4-16	Register A, Index 0Ah; bit 7 description	This bit is provided for use by software that needs to modify the time, calendar or alarm registers in the real time clock. When this bit reads back ‘1b’, these internal registers are unavailable for access by software since internal RTC logic is using them. When this bit reads back ‘0b’, software will have a guaranteed minimum window of 244 µs in which modifications to these registers are allowed... The time, calendar, and alarm information in RAM is fully available to the program when the UIP bit is 0, it is not in transition...	This bit is provided for use by software that needs to either read or write the time, calendar, or alarm registers in the real time clock. When this bit reads back ‘1b’, these internal registers are unavailable for access by software because internal RTC logic is using them. When this bit reads back ‘0b’, software will have a guaranteed minimum window of 244 µs in which reads or writes to these registers are guaranteed to be valid... The time, calendar, and alarm information in RAM is fully available to the program when the UIP bit is 0 because an update cycle is not in progress...	Rewrite to clarify meaning. The terms “modify” and “modifications” are misleading. Both reads and writes are affected.
	Register A, Index 0Ah; bits 6–4 description	0 1 0 = ... 1 1 X = ... All other values = Programming DV2-DV0 to any value except ‘010b’ or ‘11Xb’ turns the oscillator off.	0 1 0 = ... 1 1 X = ... All other values = Programming DV2–DV0 to any value except ‘010b’ or ‘11Xb’ disables the input clock from the oscillator circuit.	Clarify.

Table 1. Corrections to the *Élan™SC400 Microcontroller Register Set Reference Manual* (Continued)

Page	Item	Original Text	Change To	Comment
Note: *The following chapters in the register set manual apply only to the ÉlanSC400 microcontroller.*				
Chapter 5: Graphics Controller Indexed Registers				
5-4	CGA/MDA Index Register, Address 03x4h; diagram, R/W designation	W	R/W	Register is read/ write.
5-5	CGA/MDA Data Port, Address 03x5h; diagram, R/W designation	W	R/W	Register is read/ write.
5-7	Cursor End Register, Index 0Ah; bits 7–5 in diagram	CUR_END[4–0]	Reserved	Bit name designators are reversed in the diagram.
	Cursor End Register, Index 0Ah; bits 4–0 in diagram	Reserved	CUR_END[4–0]	
5-18	Non-display Lines Register, Index 34h; bits 1–0 description	These bits allow 2 or 4 additional non-display lines to be added...	These bits allow 1, 2, or 4 additional non-display lines to be added...	Correction.
5-20	Overflow Register, Index 36h; bit 1 description	**Vertical Display Enable End Bit 8** This is the eighth bit of the Vertical Display Enable End Register. See graphics index 37h for more detail.	**Vertical Display End Bit 8** This is the eighth bit of the Vertical Display End Register. See graphics index 37h for more detail.	Correction. Delete “Enable” (two places).
5-21	Vertical Display End Register, Index 37h; bits 7–0 description	Determines the number of the last character line to be output from the frame buffer... Value = (number of horizontal displayed character lines from memory) - 1.	Determines the number of the last character line to be output from the frame buffer... Value = (number of vertical displayed character lines from memory) – 1.	Correction. Formula is for vertical lines.
5-22	Vertical Border End Register, Index 38h; bits 7–0 description	Determines the last character line to be output to the panel at the bottom of the display. Value = (number of horizontal displayed character lines) - 1...	Determines the last character line to be output to the panel at the bottom of the display. Value = (number of vertical displayed character lines) – 1...	Correction. Formula is for vertical lines.
5-36	Pixel Clock Control Register, Index 4Ch; bits 4–3 description	Divides the base dot clock (as controlled by bits 2–0 above)...	Divides the base dot clock (as controlled by bits 2–0)...	Correction.

Table 1. Corrections to the *Élan™SC400 Microcontroller Register Set Reference Manual* (Continued)

Page	Item	Original Text	Change To	Comment
Chapter 6: PC Card Controller Indexed Registers				
6-8	Interface Status Register, Index 01h/ 41h; default bit values in diagram	0 0 x x x x x x	0 0 ? ? ? ? ? ?	"x" = non-deterministic. "?" = depends on input state.
	Interface Status Register, Index 01h/ 41h; descriptions for bits 5, 4, and 1–0	$\overline{\text{RDY\_x}}$ pin $\overline{\text{WP\_x}}$ pin $\overline{\text{BVD2\_x}}$ pin (2 places) $\overline{\text{BVD1\_x}}$ pin (2 places)	RDY_x pin WP_x pin BVD2_x pin BVD1_x pin	These pins are active high.
6-11	Interrupt and General Control Register, Index 03h/43h; bits 3–0 description, IRQ6	0 1 1 0 = IRQ6 enabled	0 1 1 0 = IRQ6 enabled Added for PC Card based floppy disk drives that support PC/AT-Compatible IRQs and DMA. An IRQ6 driven from the PCMCIA controller will not cause any PMU events (activity).	Clarification.
6-15	Address Window Enable Register, Index 06h/46h; Programming Notes text	(All text before table in programming notes section.)	When the ÉlanSC400 microcontroller is operating in Enhanced mode (see CSC index F1h[0], the five memory windows that belong to PC Card Socket B have the same meaning and controls as they do on a standard 82356 controller. When operating in Standard mode, PC Card controller Socket B memory windows 1–5 are redefined to be MMS Windows C–F. In this mode, the MMS Windows C–F base addresses and offsets are configured using the same PC Card controller registers that would have been used if the windows were configured to be PC Card windows. However, the destination devices (and other attributes that apply only to MMS Windows) are now controlled by CSC index 30h and 31h.	Rewrite of programming notes text.
	Address Window Enable Register, Index 06h/46h; Programming Notes table, center column	**MMS Window**	**MMS Window**	Correction.
		None	None	
		MMS 0	MMS C	
		MMS 1	MMS D	
		MMS 2	MMS E	
		MMS 3	MMS F	
6-56	Command Timing 0 Register, Index 3Bh; default value for bits 3–1 in diagram	0 0 0 0 1 1 1 1	0 0 0 0 0 0 0 1	Correction.

Table 1. Corrections to the *Élan™SC400 Microcontroller Register Set Reference Manual* (Continued)

Page	Item	Original Text	Change To	Comment
6-59	Command Timing 1 Register, Index 3Eh; default value for bits 3–0 in diagram	0 0 0 0 2 2 2 2	0 0 0 0 0 0 1 0	Correction.
6-62	Command Timing 2 Register, Index 7Bh; default value for bits 3–0 in diagram	0 0 0 0 3 3 3 3	0 0 0 0 0 0 1 1	Correction.

CHANGES FOR THE ÉlanSC410 MICROCONTROLLER

This section gives an overview of differences between the ÉlanSC400 microcontroller and the ÉlanSC410 microcontroller.

TheÉlanSC410 microcontroller is identical to the ÉlanSC400 microcontroller, except that the following on-chip peripheral functions are not supported in the ÉlanSC410 microcontroller:

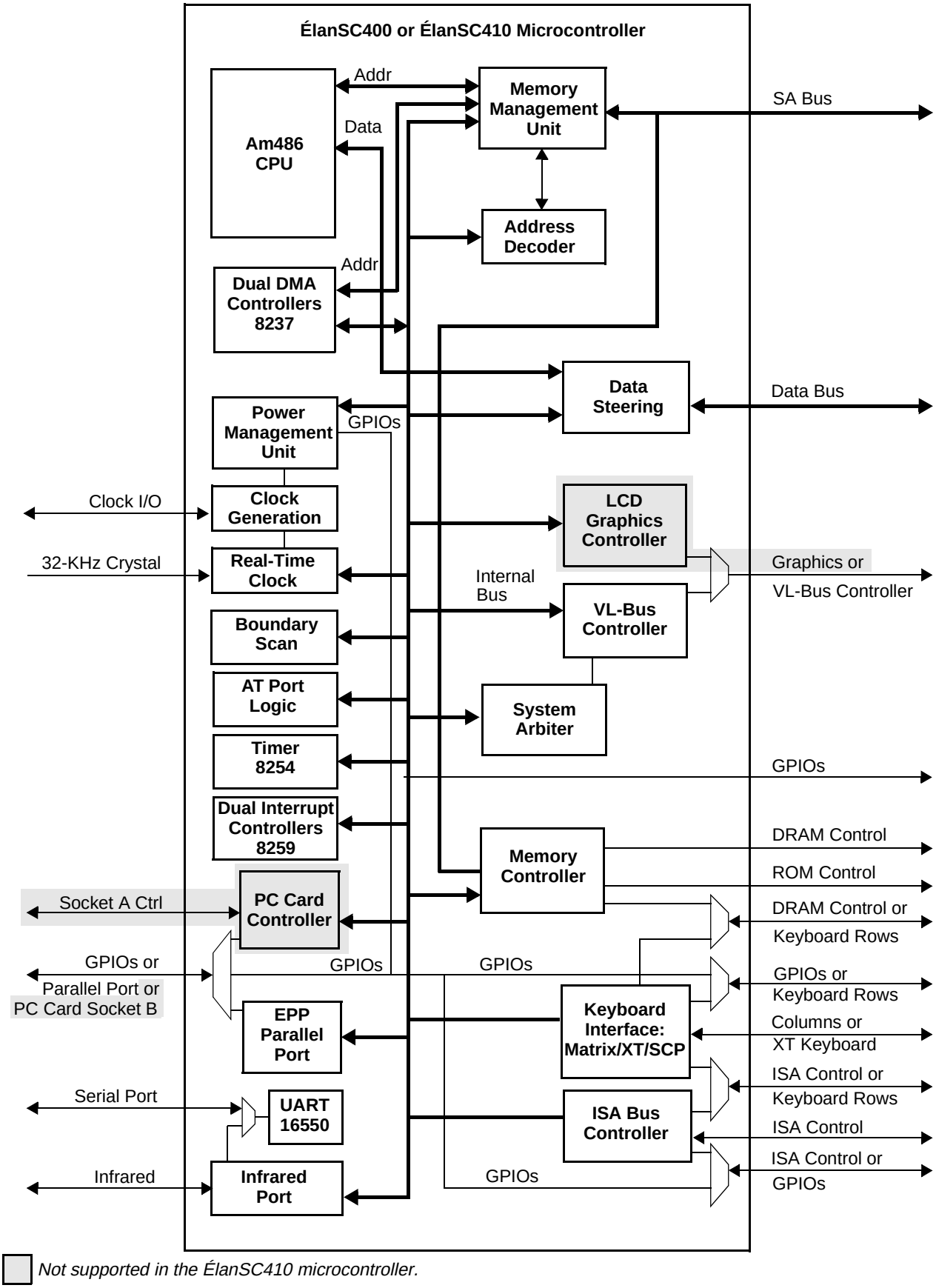
- PC Card Controller
- LCD Graphics Controller

These functions are highlighted in grey in the block diagram shown in Figure 1 on the following page.

Table 2 on page 27 lists all affected register bit fields and describes how those fields differ in the ÉlanSC410 microcontroller from the function described in the *Élan™SC400 Microcontroller Register Set Reference Manual*, order #21032A).

In general, register bits or bit field values that apply to the PC Card controller or LCD Graphics controller are reserved in the ÉlanSC410 microcontroller. The state of reserved bits must be preserved by software during read/modify/write operations.

Two PC Card controller register bits are shared by MMS Windows C–F, which is supported by the ÉlanSC410 microcontroller. Before MMS Windows C–F can be configured, software must set CSC Index bit D0h[1] = 1 and bit F1h[0] = 0.



Not supported in the ÉlanSC410 microcontroller.

Figure 1. Block Diagram—ÉlanSC400 and ÉlanSC410 Microcontrollers

Register Set Differences

The following table lists changes to the register descriptions found in the *Élan™SC400 Microcontroller Register Set Reference Manual*, order #21032A, that are required for the ÉlanSC410 microcontroller. Because the PC Card controller and LCD graphics controller are not supported on the ÉlanSC410 microcontroller, most register bit fields that control those functions are reserved. A few register bits are shared between unsupported and supported functions. These shared bit functions are also explained in the table.

Note:
The state of reserved register bits must be preserved by software during read/modify/write operations.

Page numbers in the following table refer to the *Élan™SC400 Microcontroller Register Set Reference Manual*, order #21032A.

Table 2. Register Set Manual Differences for ÉlanSC410 Microcontroller

Register	I/O Address	Bits	Comment	Page
Direct-Mapped Registers				
MDA/HGA Index Register	03B4h	7–0	Reserved on the ÉlanSC410 microcontroller	2-130
MDA/HGA Data Register	03B5	7–0	Reserved on the ÉlanSC410 microcontroller	2-131
MDA/HGA Mode Control Register	03B8h	7–0	Reserved on the ÉlanSC410 microcontroller	2-132
MDA/HGA Status Register	03BAh	7–0	Reserved on the ÉlanSC410 microcontroller	2-133
HGA Configuration Register	03BFh	7–0	Reserved on the ÉlanSC410 microcontroller	2-134
CGA Index Register	03D4h	7–0	Reserved on the ÉlanSC410 microcontroller	2-135
CGA Data Port	03D5h	7–0	Reserved on the ÉlanSC410 microcontroller	2-136
CGA Mode Control Register	03D8h	7–0	Reserved on the ÉlanSC410 microcontroller	2-137
CGA Color Select Register	03D9h	7–0	Reserved on the ÉlanSC410 microcontroller	2-138
CGA Status Register	03DAh	7–0	Reserved on the ÉlanSC410 microcontroller	2-139
Primary 82365-Compatible PC Card Controller Index Register	03E0h	7–0	Reserved on the ÉlanSC410 microcontroller	2-140
Primary 82365-Compatible PC Card Controller Data Port	03E1h	7–0	Reserved on the ÉlanSC410 microcontroller	2-141
Chip Setup and Control (CSC) Index Registers				
Cache and VL Miscellaneous Register	22h/23h Index 14h	7	Reserved on the ÉlanSC410 microcontroller	3-23
Pin Strap Status Register	22h/23h Index 20h	2	1 = Reserved on the ÉlanSC410 microcontroller	3-25
Linear $\overline{\text{ROMCS0}}$/Shadow Register	22h/23h Index 21h	6	1 = Reserved on the ÉlanSC410 microcontroller	3-26
MMS Window C–F Attributes Register	22h/23h Index 30h	7–0	On the ÉlanSC410 microcontroller, the PC Card functions referred to on these pages are not valid. However, the Enable and Mode bits referred to are shared by MMS Windows and must be configured as described (CSC Index D0h[1] = 1 and F1h[0] = 0) to enable the configuration of MMS Windows C–F.	3-38
MMS Window C–F Device Select Register	22h/23h Index 31h	7–0		3-39
Pin Mux Register B	22h/23h Index 39h	6, 5	1 = Reserved on the ÉlanSC410 microcontroller	3-45
		1, 0	0 1 = Reserved on the ÉlanSC410 microcontroller	
Pin Mux Register C	22h/23h Index 3Ah	0	1 = Reserved on the ÉlanSC410 microcontroller	3-46

Table 2. Register Set Manual Differences for ÉlanSC410 Microcontroller (Continued)

Register	I/O Address	Bits	Comment	Page
PMU Force Mode Register	22h/23h Index 40h	5	Reserved on the ÉlanSC410 microcontroller	3-51
Wake-Up Source Enable Register D	22h/23h Index 55h	7–0	Reserved on the ÉlanSC410 microcontroller	3-62
Wake-Up Source Status Register D	22h/23h Index 59h	7–0	Reserved on the ÉlanSC410 microcontroller	3-66
Activity Source Enable Register A	22h/23h Index 62h	3–2	Reserved on the ÉlanSC410 microcontroller	3-71
Activity Source Enable Register D	22h/23h Index 65h	6–1	Reserved on the ÉlanSC410 microcontroller	3-74
Activity Source Status Register A	22h/23h Index 66h	3–2	Reserved on the ÉlanSC410 microcontroller	3-75
Activity Source Status Register D	22h/23h Index 69h	6–1	Reserved on the ÉlanSC410 microcontroller	3-78
Activity Classification Register A	22h/23h Index 6Ah	3–2	Reserved on the ÉlanSC410 microcontroller	3-79
Activity Classification Register D	22h/23h Index 6Dh	6–1	Reserved on the ÉlanSC410 microcontroller	3-82
CLK_IO Pin Output Clock Select Register	22h/23h Index 83h	3–0	0 0 1 0 = Reserved on the ÉlanSC410 microcontroller	3-91
PC Card and Keyboard SMI/ NMI Enable Register	22h/23h Index 91h	3–0	Reserved on the ÉlanSC410 microcontroller	3-95
PC Card and Keyboard SMI/ NMI Status Register	22h/23h Index 95h	3–0	Reserved on the ÉlanSC410 microcontroller	3-100
SMI/NMI Select Register	22h/23h Index 98h	5	Reserved on the ÉlanSC410 microcontroller	3-104
I/O Access SMI Enable Register A	22h/23h Index 99h	3	Reserved on the ÉlanSC410 microcontroller	3-105
I/O Access SMI Enable Register B	22h/23h Index 9Ah	4, 3	Reserved on the ÉlanSC410 microcontroller	3-106
I/O Access SMI Status Register A	22h/23h Index 9Bh	3	Reserved on the ÉlanSC410 microcontroller	3-107
I/O Access SMI Status Register B	22h/23h Index 9Ch	4, 3	Reserved on the ÉlanSC410 microcontroller	3-108
Internal I/O Device Disable/ Echo Z-Bus Configuration Register	22h/23h Index D0h	1	This bit's PC Card function is invalid on the ÉlanSC410 microcontroller. However, this bit is shared by MMS Windows and must be set (= 1) to enable MMS Windows C–F setup.	3-165
DMA Resource Channel Map Register B	22h/23h Index DCh	3–0	Reserved on the ÉlanSC410 microcontroller	3-178
Internal Graphics Control Register A	22h/23h Index DDh	7–0	Reserved on the ÉlanSC410 microcontroller	3-179
Internal Graphics Control Register B	22h/23h Index DEh	7–0	Reserved on the ÉlanSC410 microcontroller	3-180
Suspend Pin State Register A	22h/23h Index E3h	7, 6	Reserved on the ÉlanSC410 microcontroller	3-184

Table 2. Register Set Manual Differences for ÉlanSC410 Microcontroller (Continued)

Register	I/O Address	Bits	Comment	Page
Suspend Mode Pin State Override Register	22h/23h Index E5h	7, 6	Reserved on the ÉlanSC410 microcontroller	3-186
PC Card Extended Features Register	22h/23h Index F0h	7–0	Reserved on the ÉlanSC410 microcontroller	3-196
PC Card Mode and DMA Control Register	22h/23h Index F1h	7–1	Reserved on the ÉlanSC410 microcontroller	3-198
		0	This bit's PC Card function is invalid on the ÉlanSC410 microcontroller. However, this bit is shared by MMS Windows and must be cleared (= 0) to enable MMS Windows C–F setup.	
PC Card Socket A/B Input Pull-up Control Register	22h/23h Index F2h	7–0	Reserved on the ÉlanSC410 microcontroller	3-200
RTC Index Registers (No Changes)				
Graphics Index Registers				
Cursor Start Register	3x4h/3x5h Index 0Ah	7–0	Reserved on the ÉlanSC410 microcontroller	5-6
Cursor End Register	3x4h/3x5h Index 0Bh	7–0	Reserved on the ÉlanSC410 microcontroller	5-7
Start Address High Register	3x4h/3x5h Index 0Ch	7–0	Reserved on the ÉlanSC410 microcontroller	5-8
Start Address Low Register	3x4h/3x5h Index 0Dh	7–0	Reserved on the ÉlanSC410 microcontroller	5-9
Cursor Address High Register	3x4h/3x5h Index 0Eh	7–0	Reserved on the ÉlanSC410 microcontroller	5-10
Cursor Address Low Register	3x4h/3x5h Index 0Fh	7–0	Reserved on the ÉlanSC410 microcontroller	5-11
Light Pen High Register (Read Only)	3x4h/3x5h Index 10h	7–0	Reserved on the ÉlanSC410 microcontroller	5-12
Light Pen Low Register (Read Only)	3x4h/3x5h Index 11h	7–0	Reserved on the ÉlanSC410 microcontroller	5-13
Horizontal Total Register	3x4h/3x5h Index 30h	7–0	Reserved on the ÉlanSC410 microcontroller	5-14
Horizontal Display End Register	3x4h/3x5h Index 31h	7–0	Reserved on the ÉlanSC410 microcontroller	5-15
Horizontal Line Pulse Start Register	3x4/3x5 Index 32h	7–0	Reserved on the ÉlanSC410 microcontroller	5-16
Horizontal Border End Register	3x4h/3x5h Index 33h	7–0	Reserved on the ÉlanSC410 microcontroller	5-17
Non-display Lines Register	3x4h/3x5h Index 34h	7–0	Reserved on the ÉlanSC410 microcontroller	5-18
Vertical Adjust Register	3x4h/3x5h Index 35h	7–0	Reserved on the ÉlanSC410 microcontroller	5-19
Overflow Register	3x4h/3x5h Index 36h	7–0	Reserved on the ÉlanSC410 microcontroller	5-20
Vertical Display End Register	3x4h/3x5h Index 37h	7–0	Reserved on the ÉlanSC410 microcontroller	5-21

Table 2. Register Set Manual Differences for ÉlanSC410 Microcontroller (Continued)

Register	I/O Address	Bits	Comment	Page
Vertical Border End Register	3x4h/3x5h Index 38h	7–0	Reserved on the ÉlanSC410 microcontroller	5-22
Frame Sync Delay Register	3x4h/3x5h Index 39h	7–0	Reserved on the ÉlanSC410 microcontroller	5-23
Dual Scan Row Adjust Register	3x4h/3x5h Index 3Bh	7–0	Reserved on the ÉlanSC410 microcontroller	5-24
Dual Scan Offset Address High Register	3x4h/3x5h Index 3Ch	7–0	Reserved on the ÉlanSC410 microcontroller	5-25
Dual Scan Offset Address Low Register	3x4h/3x5h Index 3Dh	7–0	Reserved on the ÉlanSC410 microcontroller	5-26
Offset Register	3x4h/3x5h Index 3Eh	7–0	Reserved on the ÉlanSC410 microcontroller	5-27
Underline Location Register	3x4h/3x5h Index 3Fh	7–0	Reserved on the ÉlanSC410 microcontroller	5-28
Maximum Scan Line Register	3x4h/3x5h Index 40h	7–0	Reserved on the ÉlanSC410 microcontroller	5-29
LCD Panel AC Modulation Clock Control Register	3x4h/3x5h Index 41h	7–0	Reserved on the ÉlanSC410 microcontroller	5-30
Font Table Register	3x4h/3x5h Index 42h	7–0	Reserved on the ÉlanSC410 microcontroller	5-31
Graphics Controller Grayscale Mode Register	3x4h/3x5h Index 43h	7–0	Reserved on the ÉlanSC410 microcontroller	5-32
Graphics Controller Grayscale Remapping Registers	3x4h/3x5h Index 44–4Bh	7–0	Reserved on the ÉlanSC410 microcontroller	5-34
Pixel Clock Control Register	3x4h/3x5h Index 4Ch	7–0	Reserved on the ÉlanSC410 microcontroller	5-36
Frame Buffer Base Address Register	3x4h/3x5h Index 4Dh	7–0	Reserved on the ÉlanSC410 microcontroller	5-37
Font Buffer Base Address High Byte Register	3x4h/3x5h Index 4Eh	7–0	Reserved on the ÉlanSC410 microcontroller	5-38
Frame/Font Buffer Base Address Register Low	3x4h/3x5h Index 4Fh	7–0	Reserved on the ÉlanSC410 microcontroller	5-39
PMU Control Register 1	3x4h/3x5h Index 50h	7–0	Reserved on the ÉlanSC410 microcontroller	5-40
PMU Control Register 2	3x4h/3x5h Index 51h	7–0	Reserved on the ÉlanSC410 microcontroller	5-41
Extended Feature Control Register	3x4h/3x5h Index 52h	7–0	Reserved on the ÉlanSC410 microcontroller	5-42
PC Card Controller Indexed Registers				
Identification and Revision Register	3E0h/3E1h Index 00, 40h	7–0	Reserved on the ÉlanSC410 microcontroller	6-7
Interface Status Register	3E0h/3E1h Index 01, 41h	7–0	Reserved on the ÉlanSC410 microcontroller	6-8
Power and RESETDRV Control Register	3E0h/3E1h Index 02, 42h	7–0	Reserved on the ÉlanSC410 microcontroller	6-9

Table 2. Register Set Manual Differences for ÉlanSC410 Microcontroller (Continued)

Register	I/O Address	Bits	Comment	Page
Interrupt and General Control Register	3E0h/3E1h Index 03, 43h	7–0	Reserved on the ÉlanSC410 microcontroller	6-11
Card Status Change Register	3E0h/3E1h Index 04, 44h	7–0	Reserved on the ÉlanSC410 microcontroller	6-12
Card Status Change Interrupt Configuration Register	3E0h/3E1h Index 05, 45h	7–0	Reserved on the ÉlanSC410 microcontroller	6-13
Address Window Enable Register	3E0h/3E1h Index 06, 46h	7–0	Reserved on the ÉlanSC410 microcontroller	6-15
I/O Window Control Register	3E0h/3E1h Index 07, 47h	7–0	Reserved on the ÉlanSC410 microcontroller	6-16
I/O Window 0 Start Address Low Register	3E0h/3E1h Index 08h, 48h	7–0	Reserved on the ÉlanSC410 microcontroller	6-17
I/O Window 0 Start Address High Register	3E0h/3E1h Index 09h, 49h	7–0	Reserved on the ÉlanSC410 microcontroller	6-18
I/O Window 0 Stop Address Low Register	3E0h/3E1h Index 0Ah, 4Ah	7–0	Reserved on the ÉlanSC410 microcontroller	6-19
I/O Window 0 Stop Address High Register	3E0h/3E1h Index 0Bh, 4Bh	7–0	Reserved on the ÉlanSC410 microcontroller	6-20
I/O Window 1 Start Address Low Register	3E0h/3E1h Index 0Ch, 4Ch	7–0	Reserved on the ÉlanSC410 microcontroller	6-21
I/O Window 1 Start Address High Register	3E0h/3E1h Index 0Dh, 4Dh	7–0	Reserved on the ÉlanSC410 microcontroller	6-22
I/O Window 1 Stop Address Low Register	3E0h/3E1h Index 0Eh, 4Eh	7–0	Reserved on the ÉlanSC410 microcontroller	6-23
I/O Window 1 Stop Address High Register	3E0h/3E1h Index 0Fh, 4Fh	7–0	Reserved on the ÉlanSC410 microcontroller	6-24
Memory Window 0 Start Address Low Register	3E0h/3E1h Index 10h, 50h	7–0	Reserved on the ÉlanSC410 microcontroller	6-25
Memory Window 0 Start Address High Register	3E0h/3E1h Index 11h, 51h	7–0	Reserved on the ÉlanSC410 microcontroller	6-26
Memory Window 0 Stop Address Low Register	3E0h/3E1h Index 12h, 52h	7–0	Reserved on the ÉlanSC410 microcontroller	6-27
Memory Window 0 Stop Address High Register	3E0h/3E1h Index 13h, 53h	7–0	Reserved on the ÉlanSC410 microcontroller	6-28
Memory Window 0 Address Offset Low Register	3E0h/3E1h Index 14h, 54h	7–0	Reserved on the ÉlanSC410 microcontroller	6-29
Memory Window 0 Address Offset High Register	3E0h/3E1h Index 15h, 55h	7–0	Reserved on the ÉlanSC410 microcontroller	6-30
Memory Window 1 Start Address Low Register	3E0h/3E1h Index 18h, 58h	7–0	Reserved on the ÉlanSC410 microcontroller	6-31
Memory Window 1 Start Address High Register	3E0h/3E1h Index 19h, 59h	7–0	Reserved on the ÉlanSC410 microcontroller	6-32
Memory Window 1 Stop Address Low Register	3E0h/3E1h Index 1Ah, 5Ah	7–0	Reserved on the ÉlanSC410 microcontroller	6-33
Memory Window 1 Stop Address High Register	3E0h/3E1h Index 1Bh, 5Bh	7–0	Reserved on the ÉlanSC410 microcontroller	6-34

Table 2. Register Set Manual Differences for ÉlanSC410 Microcontroller (Continued)

Register	I/O Address	Bits	Comment	Page
Memory Window 1 Address Offset Low Register	3E0h/3E1h Index 1Ch, 5Ch	7–0	Reserved on the ÉlanSC410 microcontroller	6-35
Memory Window 1 Address Offset High Register	3E0h/3E1h Index 1Dh, 5Dh	7–0	Reserved on the ÉlanSC410 microcontroller	6-36
Memory Window 2 Start Address Low Register	3E0h/3E1h Index 20h, 60h	7–0	Reserved on the ÉlanSC410 microcontroller	6-37
Memory Window 2 Start Address High Register	3E0h/3E1h Index 21h, 61h	7–0	Reserved on the ÉlanSC410 microcontroller	6-38
Memory Window 2 Stop Address Low Register	3E0h/3E1h Index 22h, 62h	7–0	Reserved on the ÉlanSC410 microcontroller	6-39
Memory Window 2 Stop Address High Register	3E0h/3E1h Index 23h, 63h	7–0	Reserved on the ÉlanSC410 microcontroller	6-40
Memory Window 2 Address Offset Low Register	3E0h/3E1h Index 24h, 64h	7–0	Reserved on the ÉlanSC410 microcontroller	6-41
Memory Window 2 Address Offset High Register	3E0h/3E1h Index 25h, 65h	7–0	Reserved on the ÉlanSC410 microcontroller	6-42
Memory Window 3 Start Address Low Register	3E0h/3E1h Index 28h, 68h	7–0	Reserved on the ÉlanSC410 microcontroller	6-43
Memory Window 3 Start Address High Register	3E0h/3E1h Index 29h, 69h	7–0	Reserved on the ÉlanSC410 microcontroller	6-44
Memory Window 3 Stop Address Low Register	3E0h/3E1h Index 2Ah, 6Ah	7–0	Reserved on the ÉlanSC410 microcontroller	6-45
Memory Window 3 Stop Address High Register	3E0h/3E1h Index 2Bh, 6Bh	7–0	Reserved on the ÉlanSC410 microcontroller	6-46
Memory Window 3 Address Offset Low Register	3E0h/3E1h Index 2Ch, 6Ch	7–0	Reserved on the ÉlanSC410 microcontroller	6-47
Memory Window 3 Address Offset High Register	3E0h/3E1h Index 2Dh, 6Dh	7–0	Reserved on the ÉlanSC410 microcontroller	6-48
Memory Window 4 Start Address Low Register	3E0h/3E1h Index 30h, 70h	7–0	Reserved on the ÉlanSC410 microcontroller	6-49
Memory Window 4 Start Address High Register	3E0h/3E1h Index 31h, 71h	7–0	Reserved on the ÉlanSC410 microcontroller	6-50
Memory Window 4 Stop Address Low Register	3E0h/3E1h Index 32h, 72h	7–0	Reserved on the ÉlanSC410 microcontroller	6-51
Memory Window 4 Stop Address High Register	3E0h/3E1h Index 33h, 73h	7–0	Reserved on the ÉlanSC410 microcontroller	6-52
Memory Window 4 Address Offset Low Register	3E0h/3E1h Index 34h, 74h	7–0	Reserved on the ÉlanSC410 microcontroller	6-53
Memory Window 4 Address Offset High Register	3E0h/3E1h Index 35h, 75h	7–0	Reserved on the ÉlanSC410 microcontroller	6-54
Setup Timing 0 Register	3E0h/3E1h Index 3Ah	7–0	Reserved on the ÉlanSC410 microcontroller	6-55
Command Timing 0 Register	3E0h/3E1h Index 3Bh	7–0	Reserved on the ÉlanSC410 microcontroller	6-56
Recovery Timing 0 Register	3E0h/3E1h Index 3Ch	7–0	Reserved on the ÉlanSC410 microcontroller	6-57

Table 2. Register Set Manual Differences for ÉlanSC410 Microcontroller (Continued)

Register	I/O Address	Bits	Comment	Page
Setup Timing 1 Register	3E0h/3E1h Index 3Dh	7–0	Reserved on the ÉlanSC410 microcontroller	6-58
Command Timing 1 Register	3E0h/3E1h Index 3Eh	7–0	Reserved on the ÉlanSC410 microcontroller	6-59
Recovery Timing 1 Register	3E0h/3E1h Index 3Fh	7–0	Reserved on the ÉlanSC410 microcontroller	6-60
Setup Timing 2 Register	3E0h/3E1h Index 7Ah	7–0	Reserved on the ÉlanSC410 microcontroller	6-61
Command Timing 2 Register	3E0h/3E1h Index 7Bh	7–0	Reserved on the ÉlanSC410 microcontroller	6-62
Recovery Timing 2 Register	3E0h/3E1h Index 7Ch	7–0	Reserved on the ÉlanSC410 microcontroller	6-63
Setup Timing 3 Register	3E0h/3E1h Index 7Dh	7–0	Reserved on the ÉlanSC410 microcontroller	6-64
Command Timing 3 Register	3E0h/3E1h Index 7Eh	7–0	Reserved on the ÉlanSC410 microcontroller	6-65
Recovery Timing 3 Register	3E0h/3E1h Index 7Fh	7–0	Reserved on the ÉlanSC410 microcontroller	6-66

INDEX

D

E

F

G

H

I

K

L

M

S

U

V

W

X