RadiSys.

RadiSys R400 Core Logic ASIC
BIOS Adaptation Guide

Version 0.9
February 10, 1997

Document Revision Record

Revision	Date	Revision Comments	Author
0.90	February 10, 1997	Preliminary Release	Tim Meese, RadiSys Corporation

Document Notes

Table of Contents

Table of Tables

Table of Figures

1. Introduction

1.1. Document Scope and Intended Audience

This document describes the software and general firmware requirements for Chipset-level BIOS support of the RadiSys R400 core logic ASIC. The scope of this document is to define the software requirements encompassing the necessary BIOS firmware in order to support the RadiSys R400 Core Logic ASIC. It is the intent of this document to do this in a sufficiently general manner, so as to be applicable to many BIOS vendor's codebases.

1.2. Overview

Chapter 1 of this document is the introduction. Chapter 2 provides an overall view of an R400-based system, and Chapter 3 delves into the R400 chipset in register-level detail providing much of the information that is necessary to provide firmware support for the R400. Chapter 4 finishes up with an overview of the System Management Mode support in the R400, and how these can be used to implement Power Management for a particular platform.

1.3. Concept

To better support embedded PC architectures; the Intel486, ULP Intel486 SX and IntelDX4 Microprocessors, and to fill the need for long lifecycle core logic components required for embedded PCs, RadiSys has developed the R400. The R400 is a single-chip ISA Core Logic ASIC. The major objectives of the R400 are flexibility, ease of use, and low system cost. The R400 incorporates directly the features needed in most or all embedded PC designs, and provides simple, low-cost, "glueless" interfaces to additional components to support added functionality, such as SVGA or PCMCIA Controllers.

1.4. Glossary of Terms

The following table contains selected terms and acronyms utilized in this document with a brief definition and section reference.

Table 1: Glossary of Terms

Term or Acronym	Definition	Related Section
FBD	Flash Boot Device: A Flash EEPROM that occupies the address space on the Local bus. This Flash EEPROM contains Video and Main BIOS code as well as BIOS extensions.	0
RFA	Resident Flash Array: One or more Flash EEPROM devices that are typically used as a mass storage device.	

1.5. Related Documents and Internet Resources

The following documents and Internet resources were referenced in this document:

RadiSys R400 Product Development Specification, Revision 3.27, RadiSys Corporation, December 11, 1996

Advanced Power Management (APM) BIOS Interface Specification, Revision 1.2, Intel Corporation and Microsoft Corporation, February 1996

Intel486™ Processor Family Programmer's Reference Manual, Intel Corporation, 1995

IntelDX4™ Processor Data Book, Intel Corporation, February 1994

Embedded Ultra-Low Power Intel486™ SX Processor Preliminary Data Sheet, Intel Corporation, September 1995

1.6. Conventions Used in this Document

Hexadecimal numbers are expressed as 0*nn*h where *nn* represents a hexadecimal number of arbitrary length. Binary numbers are expressed as 0*nn*b where *nn* represents a binary number of arbitrary length. In addition, groups of digits which are larger than four digits in length are separated with an underscore for readability (example: 0A_0000h).

Bitfield notation is used whenever square brackets are seen in an expression. A bitfield may apply to a set of signals normally considered as a bus (i.e. SA[15:0] represents the 16 least significant address lines of the processor), or to a register to denote bit positions within the register (i.e. DRAM Control Register [2:0] governs the size of the first bank of DRAM).

1.7. Feedback

Your feedback on the content, lucidity, and applicability of this document is welcome. The author of this document may be contacted via Internet e-mail at `tim.meese@radisys.com`. The RadiSys contact for ASIC Applications Engineering is Brad Reed. Brad may be reached at `brad.reed@radisys.com`. Please reference the R400 BIOS Adaptation Guide, Version 0.9.

2. R400 Core Logic Support Overview

2.1. Introduction

The functional units within the RadiSys R400 Core Logic Support ASIC are presented here from the viewpoint of the BIOS Software Engineer to serve as a framework for the BIOS chipset adaptation to support the core logic. As functional units that implement system functions, such as DRAM Control and BIOS Shadowing, are discussed in the following sections, the connection to the requisite firmware support discussed in Chapter 0 should become apparent.

Functional Units are also shown by type in Table 2. In the same way, the R400 has peripherals that are compatible with the PC architecture, and other functional units that provide core system functionality, such as DRAM Refresh Control and Power Management Support.

2.2. Functional Unit Summary

To support the standard PC architecture, the RadiSys R400 ASIC provides a Real-time Clock with battery-backed CMOS memory, a 8042-compatible Keyboard Controller, two 16550-compatible serial ports, and PS/2 Mouse Controller. The R400 Supports core system functions such as DRAM Control. The R400 includes four programmable chip select units, which may be utilized for memory regions as well as regions in I/O space. The Functional Units of each element are presented in Table 2 for easy reference.

2.3. R400 Register Summary

Pictured in Figure 1 is a summary of the R400 registers from a BIOS Developer's perspective. The **I/A** column indicates whether the number shown is an **I**ndex or an I/O Port **A**ddress. The numbers across the topmost row indicate bit positions within the registers.

The R400 register set includes an internal set of registers referred to as the "Internal" registers and also a set of registers associated with the 82C206 Megacell that is part of the R400. These register sets, and also the registers used to access the PC-Compatible peripherals, comprise the entire register set of the R400:

- 82C206 Registers
- Internal Registers
- PC-Compatible Peripherals

These register sets are demarcated in Figure 1 by the solid lines appearing before the R400 Index Register (Address 024h) and before the Keyboard Data Register (Address 060h). The first block refers to the 82C206 Macrocell Registers. The second block represents the R400 Internal registers, and the third block represents registers belonging to the R400 PC-compatible peripherals.

Table 2: Functional Unit Summary

R400 Component	Functional Unit	Description
PC Compatible Features	Interrupt Controllers (2)	8259A-Compatible Peripheral Interrupt controller.
	Timer/Counters (3)	82C54-Compatible Programmable Interval Timer with enhancements to allow remapping of peripheral addresses & interrupt assignments
	DMA Controllers (2)	8237-Compatible DMA Controller provides four 8-bit DMA channels and three 16-bit DMA channels. Internal DMA multiplexing available.
	Asynchronous Serial Ports (2)	NS16450 Compatible UART. COMB UART has support for IrDA signaling.
	Real Time Clock	Provides Motorola 146818A-compatible real time clock and alarm with 114 bytes of battery-backed CMOS memory. The internal Real-time clock is enabled or disabled by configuration pins on the R400.
	Keyboard/Mouse Controller	Implements a 8042-Compatible keyboard controller with extensions to support a PS/2 mouse, and is enabled or disabled by configuration pins on the R400.
	PC Speaker/"Port B" Functionality	
	"Port A" Functionality	Supports Port 92 Fast CPU Reset.
	IDE Interface	Supports EIDE transfer modes (PIO4). Located at address 01F0h.
Core System Support Features	DRAM Controller	Supports one or two banks of either Fast Page Mode (FPM), or Extended Data Out (EDO) memory. Total memory size ranges from 1MB to 128MB. Parity support is provided. Support for EDO/FPM type detection. Configuration options include Burst cycle, page mode, and read/write timing.
	JTAG Test Logic	Simplifies board level testing. Fully compliant with IEEE Std 1149.1-1990.
	80*x*87 Math Coprocessor Support	Provides "Clear Math Coprocessor Busy" and "Reset Math Coprocessor" functionality.
	Parity Detection Logic	Optionally store Parity Error address and assert NMI
System Management Mode (SMM) Support	HaLT detection logic	The Halt bus cycle is detected and can optionally generate an SMI
	Glitchless Clock Switching	
	Idle Timer	Two clock resolutions provide timeout values ranging from 1 millisecond to over 4 minutes. An SMI can optionally be generated on timer expiration.
	Software SMI Generation	
	External SMI Generation	EX
	STPCLK / Stop Grant protocol	
Embedded System Support Features	Programmable Chip Select Units (4)	Control for ISA Bus devices and Local bus memory regions.
	Watchdog Timer	Watchdog Timer may be programmed to assert IRQ10 or RESET on expiration. In addition, watchdog may be reloaded on 486 local bus activity or stopped when the 486 executes a HaLT instruction.

Figure 1: Programmers Register Summary

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	I/A
								Index								**22**
								Data								**23**
								IO Wait States		DMA-16 Wait States		DMA-8 Wait States		Early MEMR	DMA BCLK Divisor	01
Index																**24**
Data																**26**
0	1	0	0	0	0	0	0	0	0	0	BIOS Revision (R/W)					00
				MA[11]	MA[10]	MA[9]	MA[8]	MA[7]	MA[6]	MA[5]	MA[4]	MA[3]	MA[2]	MA[1]	MA[0]	01
			FBD Zero Wait ISA Access	Top 4MB / Top 1MB CE_BIOS#	Flash WE <1MB	BIOS WE	App WE	F8000 FFFFF	F0000 F7FFF	E8000 EFFFF	E0000 E7FFF	D8000 DFFFF	D0000 D7FFF	C8000 CFFFF	C0000 C7FFF	02
D4000 D7FFF		D0000 D3FFF		CC000 CFFFF		C8000 CBFFF		C4000 C7FFF		C0000 C3FFF		B0000 BFFFF		A0000 AFFFF		03
F8000 FFFFF		F0000 F7FFF		EC000 EFFFF		E8000 EBFFF		E4000 E7FFF		E0000 E3FFF		DC000 DFFFF		D8000 DBFFF		04
486 Byte Enable 3	486 Byte Enable 2	486 Byte Enable 1	486 Byte Enable 0	R400 SMIACT	A[26]	A[25]	A[24]	A[23]	A[22]	A[21]	A[20]	A[19]	A[18]	A[17]	A[16]	07
A[15]	A[14]	A[13]	A[12]	A[11]	A[10]	A[9]	A[8]	A[7]	A[6]	A[5]	A[4]	A[3]	A[2]	486 HLDA	Parity Missed	08
									EIDE Mode (Default: PIO Mode 0)		Parity NMI Enable	External RTC	REFRESH CSUSR1	Disable Keyboard	User RAM Bit	0A
1=16 Hz 0=1024 Hz	Reserved			Idle Timer Latch [write] Timer / Counter [read]												0B
Reserved			HALT Executed	SMI Timer Expired	Reserved	SMIINP Asserted	Top 256K DRAM access when SMIACT	A0000-BFFFF when SMIACT (override Shad)	PCS0 Resets SMI Activity Timer	Global SMI Event Signal Enable	Software SMI Event Trigger	HALT bus cycle generates SMI Event	Enable SMI Activity Timer & Event	SMIINP resets SMI Activity Timer	SMIINP causes SMI Event	0C
Chip Select Decode Mode			OR CS(*x+1*)	ISA IOW Required	ISA IOR Required	Compare and Mask Bits										0D
Chip Select Decode Mode			OR CS(*x+1*)	ISA IOW Required	ISA IOR Required	Compare and Mask Bits										0E
Chip Select Decode Mode			OR CS(*x+1*)	ISA IOW Required	ISA IOR Required	Compare and Mask Bits										0F
Chip Select Decode Mode			OR CS(*x+1*)	ISA IOW Required	ISA IOR Required	Compare and Mask Bits										10
						Reserved	HALT Executed	Reserved	Programmable Clock Reconnection Delay		HALT Freq Switch	ISABus BCLK Divisor		CLKOUT Divisor		12
				Save Parity Address	Test Parity	Addresses A0000-FFFFF Cacheable	All Other Addresses Cacheable	Bank B 2nd Module Present	Bank B Size Bits			Bank A 2nd Module Present	Bank A Size Bits			13
Reserved		Burst Mode / EDO / Page Mode Select		Addresses A0000-FFFFF LDEV Ignore	All other Addresses LDEV Ignore	Refresh Cycle Timing		ISA Refresh Enable	RAS Precharge Minimum Time	CAS to RAS Precharge Time	Address Setup before RAS	Address Hold after RAS	DRAM Burst Write Timing	DRAM Type and Burst Read Timing		14
1=16 Hz 0=1024 Hz	Reserved			Watchdog Timer Latch [write] Timer / Counter [read]												15
								Reserved	Reserved	Reserved	Reserved	IRQ10 on WD Expire Enable	Hardware Reset on WD Expire Enable	Reload WD on 486 ADS Detect	Stop WD on CPU HALT	16
								A[31]	Reserved				A[26]	A[25]	A[24]	17
								Reserved		DRQBENC			DRQAENC			18
							COM Clock Source	Disable COM1	COM Ports Pin Allocation / Port Configuration		COMCLK Divisor Value					19
								Keyboard Data								**60**
								Parity Check	0	Timer Counter 2 Out Status	Refresh Toggle	User RAM Bit	DRAM Parity Check Enable	Speaker Data Enable	Timer Counter 2 Gate Enable	**61**
								Keyboard Status / Keyboard Command								**64**
								NMI Disable	RTC / CMOS Index							**70**
								RTC / CMOS Data								**71**
								Reserved						A[20] Enable	Assert SRESET	**92**

All of the registers in the above figure are described below. The registers in boldface type in Figure 1 are shown in the "Index/Data or Peripheral Register" column in Table 3. These represent registers that are either Peripheral registers or Index/Data register pairs. The rest of the numbers, which are shown in normal typeface, are indexed registers that use the topmost pair of Index/Data registers. For example, in the table below, the 82C206 Configuration Data Register, which has Index 01 according to Table 3, has as it's access mechanism the Index/Data register pair at addresses 22h and 23h (82C206 Index Register and 82C206 Data Register, respectively) in the IO Address Space.

Table 3: Programmers Register Summary Register Descriptions

Index/Data or Peripheral Register	Index	Register Description	Index/Data or Peripheral Register	Index	Register Description
22		82C206 Index Register		0F	Chip Select 2: Compare/Mask
23		82C206 Data Register		10	Chip Select 3: Compare/Mask
	01	82C206 Configuration Data Register		12	Clock/Reset Control Register
24		Master Chipset Index Register		13	DRAM Control Register
26		Master Chipset Data Register		14	DRAM Timing Register
	00	Identification Register		15	Watchdog Timer
	01	Powerup Options Register		16	Watchdog Control
	02	BIOS Control Register		17	DMA Extension Register
	03	Shadow 1 Register		18	DMA Pin Register
	04	Shadow 2 Register		19	COM Control Register
	07	Parity Address High Register	60		Keyboard Data
	08	Parity Address Low Register	61		PortB Register
	0A	Misc Pin Configuration Register	64		Keyboard Status/Command Register
	0B	SMI Timer Control Register	70		RTC / CMOS Index Register
	0C	SMI Control/Status Register	71		RTC / CMOS Data Register
	0D	Chip Select 0: Compare/Mask	92		PortA Register
	0E	Chip Select 1: Compare/Mask			

2.4. PC-Compatible Peripherals

The R400 utilizes an 82C206 megacell to implement many standard PC system-level peripherals. They include the Real-time clock, Timer/Counter, and DMA Controller. A list of PC-Compatible registers is given in Table 4:

Table 4: PC-compatible Peripheral Registers

I/O Address Range	R400 PC-compatible Peripheral
00h - 0Fh	DMA Controller 0
20h, 21h	Master PIC
40h - 43h	Timer/Counter
60h, 64h	Keyboard Controller Data/Control
70h, 71h	Real-time Clock/NMI Mask bit
80h - 8Fh	DMA Page Registers
92h	PortA
A0h, A1h	Slave PIC
C0h - DFh	DMA Controller 1
F0h	Clear Math Coprocessor busy
F1h	Reset Math Coprocessor

PC-Compatible Keyboard data and control registers are provided at the standard addresses: 060h for the data register and 064h for the control register. The PortB (Speaker) port is provided at address 061h. and the CMOS memory is accessed in an index/data fashion also with index and data registers at 070h and 071h respectively.

2.5. BIOS Pin Configuration

In addition to the MA[11:0] signals that are latched by the R400 at power-up, there are pin configurations on the R400 that can be configured by software. These pin configurations fall into three major categories: the Interrupt Request signals (IRQs), DMA Pins, and Internal Serial Port Pins. The different pin configurations and their required firmware chipset manipulation are described below.

2.5.1. IRQ Routing

Fourteen interrupts are available external to the R400. Three selectively routable IRQ inputs are shown below in Table 5. These interrupts may be optionally routed from external peripherals by modifying the registers noted in the table.

Table 5: Selectively Routable IRQs

IRQ	IRQ Source	IRQ Routing Register	Register Index	Register Bitfield
IRQ1	Internal Keyboard Controller	Miscellaneous Pin Configuration Register	0Ah	[1]
IRQ8	Internal Real-time Clock	Miscellaneous Pin Configuration Register	0Ah	[3]
IRQ10	Internal Watchdog Timer	Watchdog Control Register	16h	[3]

Also note that the two interrupts associated with the on-board COM Ports of the R400 can be internally tri-stated and thus made shareable to external devices. This is accomplished through the use of the Enable IRQ bit in the Modem Control Register of the 16550 UART implementation. The Enable IRQ bit is bit [3] within the MCR, and the interrupts will be made shareable if this bit is clear. The location of the MCR register for each R400 Internal COM port is noted in Table 6.

Table 6: Internal COM Port IRQ Control

R400 On-board COM Port	IRQ	Modem Control Register Address
COM A	IRQ4	3FCh
COM B	IRQ3	2FCh

2.5.2. DMA Pin Binding

The DMA Pin Register can be utilized to bind the DACKA/DRQA and DACKB/DRQB external pins to a logical DMA channel within the R400 if the MA[4] Configuration Pin is set to a logic 0 on power-up. In this case, the three bit value written to the DMA Pin Register[5:3] and DMA Pin Register[2:0] corresponds to the number of the DMA channel bound internally by the R400 to the DACKB/DRQB and DACKA/DRQA signals respectively.

2.5.3. Internal Serial Port Pin Configuration

Several options are available with regard to which serial port signals are made available externally on the pins of the R400. The MA[8] Configuration pin is latched at power-up by the R400, and the value of the signal on this pin is assigned to both bits of the Pin Encoding field [6:5] in the COM Control Register. The values of the Pin Encoding field and corresponding serial port configurations are given in Table 7.

COM A UART Registers appear in the I/O Address space at 3F8h-3FFh, and COM B UART Registers appear in the I/O Address space at 2F8h-2FFh.

Table 7: Serial Port Pin Configurations

COM Control Register PINENC Field	MA[8] Pin Value	Serial Port A Configuration	Serial Port B Configuration
00b	0b	Full 8-signal Configuration: RxD, TxD, CTS, RTS, DSR, DTR, DCD, RI	Disabled
01b	(1)	4-signal Configuration: RxD, TxD, CTS, RTS	4-signal Configuration: RxD, TxD, CTS, RTS
10b	(1)	6-signal Configuration: RxD, TxD, CTS, RTS, DSR, DTR	IrDA Configuration: Ir-RxD, Ir-TxD
11b	1b	6-signal Configuration: RxD, TxD, CTS, RTS, DSR, DTR	2-signal Configuration: TxD, RxD

(1) Configuration must be set by firmware -- not a strappable option utilizing Power-up Configuration pins

2.6. Internal Register Configuration

The R400 Internal Register set provides control for core system support functions, such as BIOS ROM Decode and Control, Shadowing Operation, SMM (System Management Mode) function control, Chip Select Unit control, Clock/Reset Control, and DRAM Configuration.

The R400 Internal registers are accessed in an index/data fashion through an index and data register at addresses 024h and 026h respectively. The register set is described below in terms of the different functional units of the R400 that the Internal registers configure.

2.6.1. ID Register

The Upper 12 bits of the ID Register (R400 Internal Registers, Index 00h) are set to 400h on power up. The lower order nibble can be utilized by the BIOS to indicate revision information. These bits are reset to `0h` upon power-up.

2.6.2. Powerup Options

On power-up, the twelve least significant address lines, MA[11:0] are sampled by the R400 to determine system configuration options. The Power-up Options Register is a read-only register (R400 Internal Registers, Index 01h) that provides access to the power-up configuration bits. This register should not be initialized or written to by the BIOS. MA[10] and MA[11] are reserved for firmware usage, and can be utilized by the hardware designer to communicate information to the BIOS or Boot-block level code, such as to begin a Flash BIOS recovery process, or to clear CMOS memory. By convention, RadiSys R400-based systems utilize MA[10] to signal a forced update of the Flash Boot Device, and MA[11] to initiate a Manufacturing diagnostics loop.

2.6.3. BIOS Control Register

This register (R400 Internal Registers, Index 02h) is used to control access to the Flash Boot Device. It is used to specify when the CE_BIOS chip select is asserted and thus what regions of the FBD are made accessible to ISA memory or on the processor's local bus. It should be noted that the access options specified in the BIOS Control Register do not affect the accessibility of the FBD alias at the top of the processors address space.

Also specified are the number of wait states inserted into a FBD read access. Advanced write-enable capability for use with boot block Flash is also included.

The last bit of the BIOS Control Register specifies whether ISABus accesses to the Flash Boot Device should be performed with no wait states.

2.6.4. Shadow Control Registers

The Shadow Control Registers (R400 Internal Registers, Index 03h-04h) are two registers that control access to both the ISA Memory Space and the Upper Memory Blocks above A_0000h in DRAM. This allows the BIOS to copy code from devices residing in ISA address space, such as a Flash Boot Device or Option ROM and store the code in DRAM. Four types of access are available, and thus each region requires a two bit encoding. The four types of access are detailed in the table below.

Table 8: Shadow Register Encoding

R400 Shadow Register Encoding	Operational Characteristics
00	Read ISA Bus Write ISA Bus
01	Read ISA Bus Write DRAM
10	Read-only DRAM Write generates no ISABus or DRAM cycles
11	Read DRAM Write DRAM

Regions in the memory space above 1MB (A_0000h - F_FFFFh) vary in the degree of granularity that is available to specify access options for that region. The regions of the address space beginning with A_0000 and B_0000 have the least granularity because they are normally exclusively utilized for either VGA memory or SMRAM. The last 64K below 1MB, or the region beginning at F_0000h is divided into two regions, and has the second highest degree of granularity. The regions of memory beginning at C_0000h, D_0000h, and E_0000h have the highest degree of granularity because Option ROMs, which can be 16 Kbytes in size or smaller, are typically located here.

A summary of the granularity of memory regions that are controlled by the Shadow Control registers is presented in Table 9 below.

Table 9: Shadow Region Granularity

Shadow DRAM Region Start Address	Shadow Control Register Granularity (Kbytes)	Shadowable Regions	Comments
A_0000h	64	1	
B_0000h	64	1	
C_0000h	16	4	Typically used for Video BIOS
D_0000h	16	4	Typically used for Option ROMs or PCMCIA Real-mode Window
E_0000h	16	4	Alternate Video BIOS System BIOS
F_0000h	32	2	System BIOS

The power-on default for all regions is shaded in the table above, Read ISA Bus / Write ISA Bus, which allows access to the FBD upon power-on.

2.6.5. Parity Address Registers

The Parity Address registers (R400 Internal Registers, Index 07h-08h) are two read-only registers which are updated when a parity error occurs, and bit 11 of the DRAM Control Register (Save

Parity Address) is set. The information that is preserved by the Parity Address Registers is given in Table 10 below.

Table 10: Parity Error Saved Data

R400 Parity Address Register	R400 Register Index	Register Bitfield	Parity Error Data Description
High	07h	[15:12]	Byte Enable signals BE0 - BE3 from 486 processor
High	07h	[11]	SMIACT signal from the R400
High	07h	[10:0]	Address bits A16 through A26
Low	08h	[15:2]	Address bits A2 through A15
Low	08h	[1]	HLDA signal from 486 processor

The least significant bit of the Parity Address Low register, the PM bit, is set of a parity error occurs, but the parity error address is not stored into the Parity Address registers.

It is recommended that if a parity error does occur, that an NMI is generated so the SPA bit can be cleared and the address that generated the first parity error is preserved. This can be implemented by having a parity error generate a non-maskable interrupt (NMI) by setting bit 4 (the ENNMI bit) in the Miscellaneous Pin Configuration register. An NMI handler can then be written to clear the SPA bit, or possibly save the address that generated the parity error. The saved parity data can be provided to the user either by the BIOS user interface or the OS as diagnostic information on the next successful boot of the system.

It is also worth noting that the bootstrap configuration pins MA5 and MA6 must both be logic 1 to enable parity detection within the R400.

2.6.6. Pin Configuration Register

The Miscellaneous Pin Configuration register (R400 Internal Registers, Index 0Ah) was first mentioned in section 0, when IRQ routing was discussed. This is because the Miscellaneous Pin Configuration register contains two bits, bit 1 and bit 3 that control whether the R400's internal Keyboard controller and RTC respectively are enabled.

Bit 2 of the Miscellaneous Pin Configuration register deals with Pin Multiplexing. This bit, when clear, enables ISA Refresh capability by binding the R400 REFRESH# signal to an external pin. When set, the output of Programmable Chip Select 1, CS_USR1# , is bound to an output pin. If the Chip Select function is selected, then the ISA Refresh Enable, bit [7] in the DRAM Timing Register, must be cleared.

Bit 4 of the Miscellaneous Pin Configuration register provides for signaling of a Non-Maskable Interrupt (NMI) on a parity error. See section 0 for more information on parity exception trapping.

Lastly, bits [6:5] of the Miscellaneous Pin Configuration register represent the current EIDE mode of the R400's Internal EIDE Controller. See section 0 for more information on EIDE Mode Configuration and the R400.

2.6.7. System Management Mode Support Registers

System Management Mode (SMM) Support Registers on the R400 consist of a SMI Timer Control Register and a SMI Control/Status Register.

The SMI Timer Control Register controls a 12-bit timer. Specifically, the least significant 12 bits of the SMI Timer Control Register act as a Idle Timer latch. When written to, the Idle Timer latch value is set and written through to the 12-bit timer.

The SMI Control/Status register governs means of entry into System Management Mode through the System Management Interrupt, and also the location of SMRAM, which is the separate memory space that is utilized when the processor is in System Management Mode. Software triggerable SMI generation is supported.

This functionality is useful for implementing Power Management functionality which will be discussed in greater depth in Section 0.

2.6.8. Programmable Chip Select Registers

The R400 includes four Programmable Chip Selects (PCSs). Each PCS has a corresponding Compare/Mask register. The lower order 10 bits of each Compare/Mask register are compared based on an operational mode specified in the Compare/Mask register as well. If the comparison is true, based on the constraints specified in the MODE field of the register, then the corresponding chip select will be asserted.

The modes of access that are supported and selected via the Chip Select Decode Mode bits in each programmable chip select register are described below in Table 11.

Table 11: Programmable Chip Select Decode Modes

Mode	Mode	I/O	Memory	Description
0	000b			Chip Select Disabled
1	001b	■		Byte-addressable locations
2	010b	■		Word-addressable locations
3	011b	■		Word-addressable locations with A[15] high and A[14:10] low
4	100b		■	Word-addressable locations
5	101b		■	Byte-addressable locations using A[15:4] only
6	110b		■	Same as mode 2 above, except assert IOCS16#
7	111b		■	Same as mode 3 above, except assert IOCS16#

Required BIOS-level support is dependent upon whether ISA Bus devices or Local bus memory devices are present, and if the devices have or do not have an address decoding mechanism. Thus, PCS values will obviously be very tightly tied to the underlying hardware implementation.

As a general example, the current R400 Reference Design uses PCS2 and PCS3 for the Primary and Secondary IDE chip selects respectively. Thus the values put into the PCS2 (41F3h) and PCS3 (43F6h) registers specify word-addressable I/O ranges corresponding to 1F0-1F7h and 3F6-3F7h respectively.

2.6.9. Clock/Reset Control Register

While most registers are not inherently dependent on the core bus speed of the processor, the Clock/Reset register The values for Register 012h varies with the CLK2 value which is input to the R400. The values of Register 012h which correspond to particular values of CLK2 are shown below in Table 12.

Table 12: Clock/Reset Control Register Clock Divisor Values

Clk2Osc Speed (MHz)	CDIV R400 Clk2Osc Divider	80486 External Clock Speed (MHz)	SDIV R400 ISABus Divider	ISA Bus Speed (MHz)	Register 12 Value	Notes
100	2	50	6	8.333	0008h	Default R400 Power-up State
100	4	25	3	8.333	0002h	
100	16	6.250	2	3.125	000Fh	ISABus clock slower than 8 MHz
66	2	33	4	8.250	0004h	
66	4	16.500	2	8.250	000Eh	
66	16	4.125	2	2.063	000Fh	ISABus clock slower than 8 MHz

2.6.10. DRAM Control and Timing Registers

The DRAM Refresh Controller on the R400 is controlled by two registers -- the DRAM Control Register and the DRAM Timing Register. Fundamentally, The DRAM Control Register specifies on a per-bank basis the size of the DRAM that is contained in either of the two banks that the R400 supports. The DRAM Control Register also contains extra bits for performing address remapping of the DRAM, which is related to the use of DRAM Flash Memory.

The DRAM Timing Register allows configuration of many DRAM timing parameters including RAS and CAS assertion widths, RAS to CAS delay, RAS & CAS precharge time, and also refresh parameters of both DRAM and the ISA Bus. Selected parameters in the DRAM Timing Control will be put under CMOS Setup Control to allow the system user to configure the system to accept a wide range of DRAM components.

The DRAM Controller also features a mechanism that allows the BIOS to determine if Fast Page Mode (FPM) or Extended Data Out (EDO) memory is present in either of the R400's two banks. This is accomplished by the use of an extended ISA Bus read cycle.

Implementation details for the use of this feature to distinguish between FPM and EDO DRAM is given in section 0.

2.6.11. DMA Extension Register

Supports DMA into system memory above the 16 MB ISA address limit by allowing the firmware programmer to specify A[26:24] and A[31]. The values that are set for these address bits in the DMA Extension Register are driven onto the Local bus during a DMA operation. These bits are cleared on reset.

2.6.12. DMA Configuration Registers

Other DMA Configuration is accomplished by reading from and writing to the registers in the 82C206 macrocell in the R400. Please see section 0 for more information on configuring the 82C206. The DMA-specific configuration parameters will be described in the following sections.

2.6.12.1. 82C206 DMA Wait States

DMA wait states can be programmed separately for either 8- or 16-bit transfers. The 8- and 16-bit transfer wait states are programmed in fields [3:2] and [5:4] of the 82C206 Configuration Data Register respectively. Typical BIOS implementations should set these values to an appropriately conservative value for the system under consideration and allow user modification through a setup utility for system tweaking.

2.6.12.2. DMA Signal Configuration

Please see section

2.7. 82C206 Peripheral Configuration for a detailed explanation of the signal parameters with regard to the DMA controller that can be specified. The DMA Clock source and the MEMR# signal assertion characteristics may be manipulated.

2.6.12.3. R400 Pin Configuration

Configuring the DMA signal groups may be done in firmware if the R400 is strapped for two channel DMA support. You may recall from section 2.5.2. DMA Pin Binding, that the R400 has two signal groups (DACKA/DRQA and DACKA/DRQA) that can be selectively bound to two logical DMA channels. This is done by specifying the DMA channel number in binary format for Channel A (DMA Pin Register[2:0]) or Channel B (DMA Pin Register[5:3]).

2.6.13. On-board Serial Port Configuration Registers

The R400 includes two 16550-Compatible UARTs. These UARTs can be configured at the BIOS level for several different pinout configurations. The different configurations and the procedure for programming the R400 for each is described in section 2.5.3. Internal Serial Port Pin Configuration. Each UART is hardwired in the chipset to respond at the standard DOS COM1 and COM2 addresses, which are 3F8h and 2F8h respectively.

The R400 gives the system designer or firmware engineer the choice of generating the serial port clock internally or taking the clock from an external input. The internal clock is used to derive the clock that the UARTs use whenever COM Control Register[8] or the MA[9] Power-on Configuration bit is clear. If this is the case, and the BIOS developer wishes to support CPUs that run at different clock rates, the COM Control Register (R400 Internal Registers, Index 02h) needs to be setup appropriately for the corresponding CLK2OSC that is being fed into the R400.

Specifically, the COMCLK Divisor Value in the bitfield [4:0] must be set correctly. Table 13 below enumerates the two most common processor local bus clock rates at which the 486 processor is run: 50 MHz and 33 MHz.

Table 13: COM Control Register CLK2OSC Divisor Values

CLK2OSC Speed (MHz)	80486 External Clock Speed (MHz)	COM Clk Clock Divider	COM Clk Speed (MHz)	COM Control Register Value	Notes
100	50	54	1.852	001Bh	
66	33	36	1.833	0012h	

2.7. 82C206 Peripheral Configuration

The 82C206 Macrocell within the R400 has one configuration register. This register is accessed in an index/data fashion with the 8-bit Index and Data registers residing at 22h and 23h respectively in the address space. The single configuration register is accessed by writing 01h to the index register then writing the configuration data to the data register. Any write to the 82C206 Data register clear the Index register. Any subsequent writes of configuration data must first be preceeded with a write of 01h to the Index register.

2.7.1. MEMR Timing

When the 82C206 Configuration Data Register[1] is set, the MEMR# signal will occur one DMA clock earlier than the standard ISA timing. The normal MEMR# leading edge is delayed by one DMA clock, and this will occur if 82C206 Configuration Data Register[1] is clear.

2.7.2. DMACLK Source

If 82C206 Configuration Data Register[0] is clear, DMACLK is derived by dividing SYSCLK by 2. If set, the undivided SYSCLK will be sourced to the internal DMA Controller.

2.7.3. Wait State Configuration

The 82C206 allows the wait states for both 8- and 16-bit DMA transfers as well as IO operations to the 82C206 functional units to be assigned by software. The encoding that is used to express the number of wait states is given in Table 14.

Table 14: 82C206 Wait State Encoding

Wait State Field Value	Number of wait states inserted by 82C206 into DMA or IO Cycles
00b	One Wait State
01b	Two Wait States
10b	Three Wait States
11b	Four Wait States

DMA wait states can be programmed separately for either 8- or 16-bit transfers. The 8- and 16-bit transfer wait states are programmed in fields [3:2] and [5:4] of the 82C206 Configuration Data Register respectively. Typical BIOS implementations should set these values to an appropriately conservative value for the system under consideration and allow user modification through a setup utility for system tweaking.

3. R400 Core Logic Support Implementation Details

3.1. Register Initialization

For the purposes of illustration, values from an actual R400 BIOS implementation have been utilized to illustrate the initialization of both the R400 Internal and 82C206 registers in this chapter. In general, the most conservative values have been provided. It is left as a decision to the BIOS programmer whether a user-accessible setup mechanism is utilized to "tune" the system with less conservative values where timing is concerned. Caution must be exercised when dealing with values that can render a system unable to boot, such as choosing faster DRAM timing than the system DRAM can support. We begin with initializing the 82C206 Megacell.

3.1.1. 82C206 Register Initialization

As mentioned in section 0, the 82C206 macrocell has only one register at index 01h. The most conservative initial value for this register is 0FCh. This register is accessed by first writing 01h to the 82C206 Index Register (22h), followed immediately with a write of the data to the 82C206 Data Register (23h). Note that each write to the Data Register resets the Index Register to 00h, and thus it must be written with a 01h for subsequent Data writes to have any effect.

The most conservative wait state values (3 wait states) for both DMA and ISA bus transactions to the 82C206 peripherals were used in this example. These values should be changed or optimized based on your particular hardware configuration. For zero wait state operation, the Register Value would be 00h.

Table 15: 82C206 Register Initialization

R400 82C206 Register Name/Function	Register Index	Power-On Reset Value	Register Value	Register Mask	Read Modify Write	Notes
82C206 Configuration Data	01	(1)	FC	FF	W	

(1) Power-on Reset value of this register is not defined

3.1.2. R400 Internal Register Initialization

Noted below, in Table 16, are sample values from the BIOS implementation for the R400 Reference Design. Note that the initialization below assumes that the design utilizes Chip Selects 2 and 3 for the Primary and Secondary IDE respectively. Also note that the power-up default value of the BIOS Control Register indicates that the Video BIOS resides in the Flash Boot Device, and thus CE_BIOS# must be activated for accesses within the range of C_0000h to C_7FFFh.

Table 16: Internal Register Initialization

R400 Internal Register Name/Function	Index	Power-On Reset Value	Register Value	Register Mask	Read Modify Write	Notes
Core Logic Revision / BIOS Revision	00	4000	4001	000F		
Powerup Options Register	01	(2)		0000		Not modified
BIOS Control Register	02	70F1		0000		Not modified

Shadow 1 Register	03	0000	0000	FFFF	W	
Shadow 2 Register	04	0000	0000	FFFF	W	
Parity Address High Register	07	(1)		0000		Not modified
Parity Address Low Register	08	(1)		0000		Not modified
Misc Pin Configuration Register	0A	0001	0000	007F	RMW	
SMI Timer Control Register	0B	0000	0000	8FFF	RMW	
SMI Control / Status Register	0C	0000	0000	1FFF	RMW	
Chip Select 0 Compare/Mask Register	0D	0000	0000	FFFF	W	
Chip Select 1 Compare/Mask Register	0E	0000	0000	FFFF	W	
Chip Select 2 Compare/Mask Register	0F	0000	41F3	FFFF	W	Primary IDE IO Addr: 1F0-1F7
Chip Select 3 Compare/Mask Register	10	0000	43F6	FFFF	W	Secondary IDE IO Addr: 3F6-3F7
Clock/Reset Control Register	12	0008	(3)	007F	RMW	
DRAM Control Register	13	0011	0011	0FFF	RMW	
DRAM Timing Register	14	0DFF	0DFF	3FFF	RMW	
Watchdog Timer Register	15	0000	0000	FFFF	W	
Watchdog Control Register	16	0000	0000	00FF	RMW	
DMA Extension Register	17	0000		00FF		Not Modified
DMA Pin Register	18	0011	0000	00FF	RMW	
COM Control Register	19	017B	0000	01FF	RMW	

(1) Power-on Reset does not affect this register
(2) Power-on Reset value of this register is not defined
(3) Register value should be determined by System core speed

3.2. System Memory Configuration

The R400 supports two banks of DRAM. Each bank is either 36 or 32 bits wide depending upon whether parity is used or not respectively. System Memory configuration routines written to support the R400 should generally perform three tasks. The first task is to determine the type of memory being utilized. This normally falls into one of two categories: Fast Page Mode (FPM) or Extended Data Out (EDO). This information is utilized in the second task, which is to determine the proper timing configuration for the particular DRAM and local bus speed. The final task is the familiar DRAM sizing, which is done by virtually all BIOS firmware. We begin with EDO/FPM Detection.

3.2.1. EDO/FPM Detection Algorithm

Standard R400 designs can utilize a hardware feature to distinguish whether the system is running with Extended Data Out (EDO) or Fast Page Mode (FPM) DRAM.

Hardware support for this feature consists of providing a pullup resistor on one of the 32 data lines on the 486 MD-bus. The R400 reference design uses MD[0] for this purpose. The BIOS then utilizes this bit, and a special operating mode of the DRAM Controller to detect the presence of EDO DRAM.

The DRAM Timing Register is used to enter and exit this detection mode. Specifically bits [13:12], whose function is described in Table 17 below, are set to 10b to enable the detection mode. After detection, they are set according to whether there is a DRAM type mismatch in the system. The algorithm for EDO/FPM Detection is shown below in Figure 2.

Table 17: DRAM Timing Register MODE Selection

MODE Bitfield	Functional Mode Description	Operating Condition
00b	• RDY# Assertion • No Burst Cycles • No Page Mode	This mode is typically used while the BIOS is running, and also if bank DRAM types (EDO or FPM) do not match if more than one bank is present. This is the

		default Power-on Reset Mode
01b	• BRDY# Assertion • Burst Cycles • No Page Mode	
10b	• RDY# Assertion • No Burst Cycles • No Page Mode • EDO/FPM Detection	This mode is used for EDO/FPM Detection.
11b	• BRDY# Assertion • Burst Cycles • Page Mode	This mode is used for Normal Operation

It should be noted that the R400 DRAM Controller will not perform memory accesses optimally

puts the following restriction on memory configurations: both banks, if populated, must contain the same type of memory (i.e. either FPM or EDO). Banks cannot both be populated with SIMMs that are not of the same memory type. If this

The following EDO/FPM Detection routine is meant to be implemented after the BIOS has performed system DRAM sizing.

It should be noted that the pseudo-code below is meant to be generic and similar to natural language. Any similarity to Ada or Pascal is strictly non-intentional. The bitfield notation is used with a twist – bitfields may be assigned to logical names which merely act as 'placeholders'. In a real implementation, a bit mask would probably be employed here to manipulate specific bits within the R400 DRAM registers. A general begin/end conditional block syntax is also utilized. The only language-specific elements are the comment line delimeter (//) and the not-equal-to operator (!=) taken from C/C++.

Figure 2: EDO/FPM Detection Algorithm

```
Begin EDO/FPM Detection procedure

//
// Declarations
//

Bitfield DRAM Control Register[7:4] is "Bank B Size bits"
```

```
Bitfield DRAM Control Register[3:0] is “Bank A Size bits”
Bitfield DRAM Timing Register[13:12] is “DRAM Mode bits”
Bitfield DRAM Timing Register[1] is “EDO/FPM Type bit”

Byte at address 0h is “Test location”
Bitfield Test location[0] is “EDO/FPM Detection bit”

Enumeration {EDO, FPM} is Type DRAM Type
Type “bank A memory type” is DRAM Type
Type “bank B memory type” is DRAM Type
Type “detected memory type” is DRAM Type

// Setup EDO/FPM Detection Mode
Set DRAM Mode bits to 10b

If (Bank A Size bits != 0), then
       Begin
       // DRAM was detected in bank A by the autosizing routine
       // Turn off bank B to isolate bank A
       Clear Bank B Size bits
       Clear Test location
       If EDO/FPM Detection bit is 0b, then bank A memory type is FPM
       If EDO/FPM Detection bit is 1b, then bank A memory type is EDO
       End

If (Bank B Size bits != 0), then
       Begin
       // DRAM was detected in bank B by the autosizing routine
       // Turn off bank A to isolate bank B
       Clear Bank B Size bits
       Clear Test location
       If EDO/FPM Detection bit is 0b, then bank B memory type is FPM
       If EDO/FPM Detection bit is 1b, then bank B memory type is EDO
       End

If ((Bank A Size bits != 0) and (Bank B Size bits != 0)), then
       Begin
       If (bank A memory type != bank B memory type), then
              Begin
              Set DRAM Mode Bits to 00b
              Set [1] of DRAM Timing Register to 1b (indicates FPM DRAM)
              Exit EDO/FPM Detection procedure
              End
       End

// Either both banks are present and of the same DRAM type at this
// point, or only one bank is present


If (Bank A Size bits != 0), then
       Begin
       detected memory type is bank A memory type
       End
else

       Begin
       detected memory type is bank B memory type
       End

If detected memory type is FPM, then
       Begin
       Set EDO/FPM Type bit to 1b // indicates FPM DRAM
       Set DRAM Mode bits to 11b
       End
```

```
If detected memory type is EDO, then
       Begin
       Set EDO/FPM Type bit to 0b // indicates EDO DRAM
       Set MODE Bits of DRAM Timing Register to 11b
       End

End EDO/FPM Detection procedure
```

3.2.2. DRAM Sizing

The R400 allows the two banks of memory to be accessed independently and the starting DRAM Address is determined by which bank is active. Thus each bank can be sized by the BIOS independently, and the resulting memory configuration enabled after sizing of each bank has been completed. Table 18 shows the values that can be specified in the size bit field for each bank. A bank can be disabled by writing 00h into the bank size field for the particular bank. Bank A's bank size field is DRAM Control Register[3:0] whereas bank B's bank size field is DRAM Control Register[7:4]. The Power-on reset value for both banks is 01h, or 1MB.

Table 18: DRAM Bank Sizing Detail

Bank Size Configuration Value	Total Bank Memory Size (Mbytes)	Module SIMM Form Factor	First Module	Second Module
00h	0	Module not present		
01h	1	256K x 32	■	
02h	2	512K x 32	■	
03h	4	1024K x 32	■	
04h	8	2048K x 32	■	
05h	16	4096K x 32	■	
06h	32	8192K x 32	■	
07h	64	16384K x 32	■	
08h	0	Module not present		
09h	2	256K x 32	■	■
0Ah	4	512K x 32	■	■
0Bh	8	1024K x 32	■	■
0Ch	16	2048K x 32	■	■
0Dh	32	4096K x 32	■	■
0Eh	64	8192K x 32	■	■
0Fh	128	16384K x 32	■	■

3.2.3. DRAM Refresh Timing

The R400 allows two different timing modes for DRAM Refresh Cycle Timing. These are specified in Table 19. Normally, the bits governing the DRAM Refresh Timing (DRAM Timing Register[9:8]) should be set to 01b. These bits can be set to 10b for Extended Refresh DRAM. Once again, the hardware configuration will dictate what value these bits take on and whether they should be user modifiable.

Table 19: DRAM Refresh Timing Parameters

DRAM Timing Register[8:9]	Refresh Period (microseconds)	Operating Mode
00b	∞	Refresh Disabled
01b	15.50	Refresh Period for Standard DRAM
10b	93.03	Refresh Period for Extended Refresh DRAM
11b		Special Test Mode

3.2.4. ISA Refresh

ISA Refresh should be enabled, i.e. DRAM Timing Register[7] set to 1b, when BIOS code is executing out of the Flash Boot Device. After BIOS code has been shadowed, this bit can be either specified by the user or tailored to a specific application. This bit is set to 1b on Power-on reset.

3.2.5. LDEV Recognition

The bitfield DRAM Timing Register[11:10] controls whether the LDEV# signal is recognized for DRAM accesses by the R400. The most significant bit, [11], of this bitfield, corresponds to the memory region A_0000h to F_FFFFh, and the second bit, [10], corresponds to the memory exclusive of the range specified by the first bit. If the bits are clear, then LDEV# is ignored by the R400 for the respective memory region. This allows local bus devices overlaying DRAM to drive LDEV#.

3.2.6. DRAM Timing Configuration

DRAM Timing on the R400 is mainly governed by two bitfields:

- DRAM Timing Register[13:12]
- DRAM Timing Register[6:0]

The first of these bitfields was discussed in section 3.2.1. EDO/FPM Detection Algorithm when EDO/FPM Detection was discussed. This is because the third mode described puts the R400 into a mode where reads are slowed down so that EDO/FPM Detection can be performed. This is the slowest memory timing mode of the R400 because no burst cycles are generated and accesses do not utilize page mode.

The mode that is selected by default on Power-on reset, is slightly faster in that read operations are not slower for EDO/FPM detection. This is the mode that should be selected by an EDO/FPM Detection algorithm if two banks of different DRAM types (i.e. EDO/FPM or FPM/EDO in BankA/BankB) are detected.

The normal operating mode, with DRAM Timing Register[13:12] set to 11b, allows burst cycle generation and page mode operation. This mode should be set when EDO/FPM determination has taken place and only one bank of a particular DRAM type is detected or two banks of the same DRAM type are detected. This is the fastest mode of operation that can be selected with these two bits.

The next bitfield that determines DRAM timing are the seven least significant bits of the same register. DRAM Timing Register[6:0] are set according to DRAM speed, the processor local bus speed, and the DRAM type. The BIOS normally determines the latter two of these three parameters during the Power On Self Test, so if DRAM configuration is requested of the user, the BIOS can require only one user-supplied parameter, DRAM speed, to calculate the optimum DRAM configuration.

Of course, any or all of the bits in DRAM Timing Register[6:0] can also be user configurable via a setup utility or optimized according to the target hardware configuration. Table 20 below gives the optimum value of DRAM Timing Register[6:0] based on the processor local bus speed, DRAM speed, and DRAM type.

Table 20: DRAM Timing Parameters

CPU Local Bus Speed (MHz)	DRAM Speed (ns)	EDO	FPM	DRAM Timing Register[0:6]	Notes
33	60	■		000 0000b	
	70	■		100 1001b	
33	60		■	000 0010b	
	70		■	100 1010b	
	80		■	111 1111b	
50	60	■		111 1001b	
50	50		■	111 1010b	
	60		■	111 1011b	

3.3. EIDE Transfer Mode

The R400 supports EIDE drives at FPIO Mode 3 or FPIO Mode 4 PIO Mode 0 is also supported to preserve support for slower drives, or when a faster FPIO Mode is not desired.

The Advanced PIO modes that a drive supports can be obtained by issuing the Get Device Information command (0ECh) to Port 1F7h for the primary drive. This command will return a 512-byte block of device-specific information on the drive via port 1F0h. The byte at offset 80h in this data block contains the Advanced PIO modes supported by the drive. From this information, and knowledge of how fast the local bus speed of the 486 is, the proper bits can be set in the Miscellaneous Pin Configuration Register[6:5]. Note that there are two entries for PIO Mode 4, of which one is valid only at a 50 MHz local bus speed.

Table 21 below describes the possible values for the Miscellaneous Pin Configuration Register[6:5] bitfield that should be used by firmware to select the FPIO Mode that the R400 will utilize in I/O operations with the drive. It should be noted that these bits are set to 00h on powerup.

Table 21: EIDE Transfer Modes

Misc Pin Config [6:5]	CLKIN Clock Constraints	PIO Mode Description
00b	Valid at any CLKIN clock rate	PIO Mode 0
01b	Valid only at 33 MHz or slower CLKIN	PIO Mode 3
10b	Optimized for 50 MHz CLKIN Transfers at less than this CLKIN will be less efficient	PIO Mode 4
11b	Valid only at 33 MHz or slower CLKIN	PIO Mode 4

4. R400 System Management Mode Support Overview

4.1. System Management Mode (SMM)

In general, many system management schemes, including power management and the management of sophisticated subsystems, such as USB, are implemented using the System Management Mode (SMM) of the Intel486 processor. SMM provides a special-purpose operating mode that is distinct and transparent to an Operating System running in Real or Protected Mode.

SMM is invoked through a special interrupt called the system management interrupt or SMI. Upon assertion of the SMI signal to the processor, the processor saves it's current context, and begins execution at a fixed offset system management RAM or SMRAM. The code that is executed in SMRAM is known as an SMM handler, which will perform system level tasks. These tasks may include responding to a timeout set by a power management routine to shut down idle system resources. The processor is now executing in system management mode (SMM), and has access to entire 486 address range. The prior processor mode, whether it be Protected, Virtual86, or Real mode may be returned to by execution of the RSM instruction by the SMM handler. Additionally, Real mode may be returned to by resetting the processor.

Resumption of normal processor operation is accomplished through execution of the Resume instruction by the SMM handler.

4.2. The System Management Interrupt (SMI)

As mentioned in the previous section, the method of entering SMM is the system management interrupt. An SMI causes the CPU to save it's current context and begin execution of an SMM handler.

4.2.1. Enabling SMI

The R400 Global SMI enable is controlled by the SMI Control/Status Register[5] bit. This bit must be set in order for an SMI to be signaled to the CPU. This bit is clear on Power-on reset.

4.2.2. Software SMI

The location of the SMM handler is partially determined by an internal register called the SMBASE register. The SMBASE register image resides at offset 7EF8h in SMRAM after the processor has saved it's context. The SMBASE register may only be manipulated in Intel486 processors by entering SMM, modifying the saved context of the processor including the SMBASE register, and executing a resume instruction, causing the processor to restore the saved context including the new SMBASE value. On the next entry into SMM, the new SMBASE value will take affect.

In order to accomplish SMM configuration tasks, such as modifying the SMBASE register, SMM firmware needs a way to signal an SMI on-demand "in software" exclusive of hardware mechanisms that are present to generate an SMI externally. This is accomplished by manipulating the R400's SMI Control Status Register. This is accomplished in the R400 by setting SMI Control/Status Register[4], which is a Software SMI Event Trigger. If SMI generation is enabled, an SMI will be sent to the processor.

4.2.3. Hardware SMI

The SMIINP# signal is available as an input to the R400. When the SMI Control/Status Register[0] bit is set, then an SMI is generated whenever SMIINP# is asserted. Note that this bit should not be set if the NMI pin multiplexing is chosen (MA[6:5] = 11b) as this pin is shared with SMIINP. On Power-on reset, this functionality is disabled (SMI Control/Status Register[0] is cleared).

4.2.4. Halt Cycle

If configured, the detection of a Halt Cycle will cause an SMI. The SMI Control/Status Register[3] bit enables an SMI when a Halt bus cycle is detected by the R400. This bit is cleared on Power-on reset.

4.2.5. SMI Timer

If the SMI Timer expires and is enabled to generate an SMI, an SMI will be set to the processor. If the SMI Global Enable is set, then if SMI Control/Status Register[2] is set also, an SMI will be signalled when the SMI Timer expires.

4.3. SMI Activity Timer

The SMI Timer Control register allows access to the value of a 12-bit countdown timer. The most significant bit of this register controls the frequency at which the clock value is decremented, and is referred to as the clock prescaler value. Writes to the timer value register use a latched write mechanism, allowing a new value to be written into the timer value register while the timer is still running. The current value of the counter can be obtained by simply reading the timer value register. Reads also return the value of the clock prescaler bit.

The SMI Activity Timer can be clocked at a 1024 Hz or 16 Hz rate. Both of these clocks are derived from the R400's 32,768 Hz clock. At 1024 Hz, the timer has a granularity of about 1 millisecond with the largest timeout value being approximately 4 seconds. At 16 Hz, the timer has a granularity of 62.5 milliseconds and the largest timeout period is 256 seconds, or just over 4 minutes.

Note that SMI Control/Status Register[2] controls whether timer expirations generate an SMI. This bit also enables the Activity Timer to count down.

4.3.1. Programmable Chip Selects

The Programmable chip select, PCS0 may be used to reset the SMI Timer to it's last latched value. This is accomplished by setting SMI Control/Status Register[6]. Because the R400 Programmable Chip Selects can be cascaded by setting [12], PCS0 can be cascaded with the other 3 chip selects provided that the Chip Select *n*[12] is set where n={0}, {0,1}, {0,1,2}, or {0,1,2,3}.

4.3.2. SMI Generation

If the SMI Timer expires and is enabled to generate an SMI, an SMI will be set to the processor. If the SMI Global Enable is set, then if SMI Control/Status Register[2] is set also, an SMI will be signalled when the SMI Timer expires.

4.3.3. Activity Timer Reset

The SMI Activity Timer can be reset to the last value that was written into the SMI Timer Control Register. This is accomplished by setting SMI Control/Status Register[2]. This causes the timer to be reset by the SMIINP# external input. The SMI Activity Timer can also be directly reset to an arbitrary value by writing the SMI Timer Control Register. Any writes to this register latch the value written and the latched data is passed through to the timer engine.

4.4. Resume (RSM) Instruction

The only way to switch from System Management Mode into the previous operating mode, short of resetting the processor, is to execute the Resume instruction. This instruction, which has the opcode 0FAAh, is accomplished in between 452 and 465 clock cycles on an Intel486 processor based on the reason for exiting SMM. Upon execution of the RSM instruction, the processor restores the original context that was saved when SMM was entered. SMM is exited, and the processor resumes execution of the interrupted application or operating system.

4.5. SMRAM Mapping

As alluded to earlier in section **4.2.1. Enabling SMI**

The R400 Global SMI enable is controlled by the SMI Control/Status Register[5] bit. This bit must be set in order for an SMI to be signaled to the CPU. This bit is clear on Power-on reset.

4.2.2. Software SMI, the location of SMRAM can be changed by changing the internal SMBASE register in Intel486 processors. The R400 also provides a mechanism that allows SMRAM to be relocated. SMRAM can be relocated by the R400 to either the top 256Kbytes of System DRAM, or the 128Kbytes between A_0000h and B_FFFFh by . Note that for accesses to the region A_0000h to B_FFFFh to be mapped to DRAM during SMM, the corresponding bits in the Shadow 1 Register, [3:0], must be set to read/write DRAM access.

4.6. CPU Clock Speed Reduction

Power that is consumed by the processor is proportional to the clock rate that the CPU runs at. Thus a way to reduce overall system power consumption is to slow down the clock going to the CPU when the system is idle. The R400 allows the BIOS firmware programmer to accomplish this in one of two ways.

- Issue a Halt (HLT) instruction with the Clock/Reset Control Register[4] bit set to glitchlessly switch the CPU clock to 32 KHz
- Change the CLKOUT Divisor in the Clock/Reset Control Register[1:0]

4.6.1. Halt (HLT) Instruction

The Halt (HLT) instruction can be utilized to switch the processor clock to 32 KHz. When this clock switch is performed, the CLKOUT0 pin is completely stopped and driven low, aiding power conservation while the CPU is halted. The Clock/Reset Control Register[4] must be set for clock switching to take place.

4.6.2. Clock Throttling

The second method of lowering the processor clock rate and lowering overall power consumption is through clock throttling. This is accomplished by changing the divisor, and thus sourcing a lower speed CLKOUT to the processor. Initially, the Clock/Reset Control Register[1:0] bitfield is 00b. If the R400 detects that either 10b or 11b is being written to the bitfield, a stop grant arbitration sequence will be initiated before the clock frequency is changed.

Table 22: CLK2OSC Divisors

Clock/Reset Control Register[1:0]	CLK2OSC Divisor	Approximate Power Consumption Ratio
00b	2	1
01b	2	1
10b	4	0.5
11b	16	0.125

4.6.3. Oscillator Reconnection Delay

Finally, the R400 Clock/Reset Control Register includes a feature to allow a delay before sourcing an oscillator or PLL clock synthesizer back to the processor. This allows these devices extra time, if needed, to return to their rated frequency after they have been powered down by the OSC_OFF# signal. After the observed delay, STPCLK# will be held for an additional 1ms before the clock is sourced back to the processor. The programmable delays are enumerated below in Table 23.

Table 23: Programmable Oscillator Reconnection Delay

Clock/Reset Control Register[6:5]	Oscillator Reconnection Delay (ms)
00b	No Delay
01b	0.5
10b	2.0
11b	8.0