



BIOS Programmer's Guide

Transmeta Crusoe™ Processor

TM5700/TM5900 Hardware Version 1.x
Code Morphing Software Version 4.5.x

Transmeta Proprietary Information Provided Under
Nondisclosure Agreement

Property of:

Transmeta Corporation
3990 Freedom Circle
Santa Clara, CA 95054
USA
(408) 919-3000
<http://www.transmeta.com>

The information contained in this document is provided solely for use in connection with Transmeta products, and Transmeta reserves all rights in and to such information and the products discussed herein. This document should not be construed as transferring or granting a license to any intellectual property rights, whether express, implied, arising through estoppel or otherwise. Except as may be agreed in writing by Transmeta, all Transmeta products are provided "as is" and without a warranty of any kind, and Transmeta hereby disclaims all warranties, express or implied, relating to Transmeta's products, including, but not limited to, the implied warranties of merchantability, fitness for a particular purpose and non-infringement of third party intellectual property. Transmeta products may contain design defects or errors which may cause the products to deviate from published specifications, and Transmeta documents may contain inaccurate information. Transmeta makes no representations or warranties with respect to the accuracy or completeness of the information contained in this document, and Transmeta reserves the right to change product descriptions and product specifications at any time, without notice.

Transmeta products have not been designed, tested, or manufactured for use in any application where failure, malfunction, or inaccuracy carries a risk of death, bodily injury, or damage to tangible property, including, but not limited to, use in factory control systems, medical devices or facilities, nuclear facilities, aircraft, watercraft or automobile navigation or communication, emergency systems, or other applications with a similar degree of potential hazard.

Transmeta reserves the right to discontinue any product or product document at any time without notice, or to change any feature or function of any Transmeta product or product document at any time without notice.

Trademarks: Transmeta, the Transmeta logo, Crusoe, the Crusoe logo, Efficeon, the Efficeon logo, Code Morphing, LongRun, and combinations thereof are trademarks of Transmeta Corporation in the USA and other countries. Other product names and brands used in this document are for identification purposes only, and are the property of their respective owners.

Copyright © 1/23/04 Transmeta Corporation. All rights reserved.

Table of Contents

	List of Tables	5
	Introduction	7
Chapter 1	Processor and Frequency Detection	9
	1.1 Processor Detection	9
	1.1.1 String Representation	9
	1.1.2 Reserved Functions	10
	1.1.3 CPUID Standard Functions	11
	1.1.4 CPUID Extended Functions	13
	1.1.5 CPUID Transmeta-Specific Functions	15
	1.2 Frequency Detection	17
Chapter 2	PCI Configuration Registers	19
	2.1 SMRAM Differences	20
	2.2 I/O Mapped Registers	20
	2.3 PCI Configuration Registers	23
	2.3.1 Host-to-PCI Bridge Registers (Function 0)	24
	2.3.2 SDRAM Memory Controller Registers (Function 1)	50
	2.3.3 Extended BIOS Scratch Pad (Function 2)	54
	2.3.4 DDR SDRAM Memory Controller Registers (Func. 3)	57
Chapter 3	Model-Specific Registers	63
	3.1 Introduction	63
	3.2 Intel-Compatible MSRs	63
	3.3 Transmeta-Specific MSRs	65
Chapter 4	Power Management	71
	4.1 Power Management Control	71
	4.1.1 Processor Power States	72
	4.1.2 State Transitions	73
	4.1.3 State Preservation	73
	4.2 Power Management Registers	74
	4.3 LongRun™ Power Management	74
	4.4 Example Configuration for PIIX4	75
	4.5 Special Considerations	75
	4.5.1 PCI Arbiter	75
	4.5.2 Configuration Register Context during STR	76
	4.5.3 DRAM Refresh	76
	4.5.4 STPCLK Behavior of ALI 1535	76
Chapter 5	System Management Mode	77

5.1	SMM Address Space.....	77
5.1.1	Standard and Extended SMM Memory Configuration.....	77
5.1.2	High SMM Memory Configuration	78
5.1.3	Address Space Location and Use.....	78
5.2	SMM Initialization and State Change	79
5.2.1	State Save Map.....	80
5.2.2	State Save Prior to Power Off	81
5.2.3	Address Space Caching.....	81
5.3	SMM Execution Environment	81
5.3.1	SMM Entry State	81
5.3.2	Exceptions, Interrupts, and A20 Masking within SMM	82
5.3.3	SMM Exit Considerations	82
Appendix A	Power Management Configuration Code Example	85
Appendix B	PCI Passive Release	89
Appendix C	Initial Processor State	91
Appendix D	BIOS and Keyboard Controller Checklists	93

List of Tables

Table 1:	String Fragment Order for Strings Returned by CPUID Functions	10
Table 2:	CPUID Standard Functions	11
Table 3:	CPUID Extended Functions	13
Table 4:	CPUID Transmeta-Specific Extended Functions	15
Table 5:	PCI Register Access Classifications	19
Table 6:	SDRAM Differences Between 82443BX and Transmeta VNB	20
Table 7:	I/O Mapped Register Summary	20
Table 8:	PCI Registers (Function 0) Summary	24
Table 9:	SDRAM Memory Controller Registers (PCI Function 1) Summary.....	50
Table 10:	Extended BIOS Scratch Pad Registers (PCI Function 2) Summary.....	54
Table 11:	DDR SDRAM Memory Controller Registers (PCI Function 3) Summary.....	57
Table 12:	Intel-Compatible MSR Summary	63
Table 13:	Transmeta-Specific MSR Summary	65
Table 14:	Power Management System States	71
Table 15:	Processor Power States	72
Table 16:	Example Settings for Power Management Registers (PIIX4 Southbridge).....	75
Table 17:	Initial Processor State	91

Introduction

This document provides information necessary to implement a system BIOS that supports the Crusoe™ TM5700/TM5900 processor family. The Crusoe processor consists of a Transmeta microprocessor and x86 Code Morphing™ Software that together provide the functionality of an x86-compatible processor.

This document focuses on unique features of the Crusoe processor and is not intended to document generic system BIOS requirements.

Reference Documents

The following documents should be used in conjunction with this guide:

- *Crusoe™ Processor Data Book*
- *Crusoe™ Processor Development and Manufacturing Guide*
- *PCI Local Bus Specification*, PCI SIG Revision 2.1
- *PCI Hardware and Software - Architecture and Design (4th ed.)*, by Edward Solari and George Willse, Annabooks, San Diego, 1998
- JEDEC Standard No. 21-C (11/25/97)

Processor and Frequency Detection

1.1 Processor Detection

The recommended method of processor detection for the Crusoe™ Processor Model TM5700/TM5900 is to use the CPUID instruction. Before executing CPUID, the ID bit in the EFLAGS register should be tested. The TM5700/TM5900 ID bit can be modified which indicates that the processor supports the CPUID instruction.

The CPUID instruction uses the EAX register as an input. If the upper 16 bits in EAX are clear (0000h), the CPUID standard function set is selected. If the upper 16 bits in EAX are set to 8000h, the extended function set is selected. A Transmeta-specific extended function set is also available by setting the upper 16 bits of EAX to 8086h. The output or return values are loaded into the EAX, EBX, ECX and EDX registers. CPUID is a non-privileged serializing instruction.

The TM5700/TM5900 supports the standard and extended functions listed in the following tables. All EAX input values other than those listed in the tables are reserved and return the value of 0000 0000h in the output registers. Implicitly reserved functions have zero return values in all output registers. Additionally, some of the bits listed are designated as “reserved” bits. Software should not rely on the value of reserved bits.

1.1.1 String Representation

Several CPUID function codes return character strings. These strings have the common properties described below.

Note
These properties apply only to Transmeta-supplied defaults. Programmable strings may deviate from this description.

Character set	For maximum compatibility, Transmeta uses only the most universal subset of ASCII for string values: the Unique Graphics Characters of ISO 646:1991 and ECMA-6 (the latter is available from http://www.ecma.ch).
Zero fill	CPUID function codes that return string data specify the size for each string. Unused characters are filled with a value of 00h.
Null termination not guaranteed	The entire reserved size of the string may be taken up with printable characters; nulls are only present if required for fill. Thus, programs should not assume these are null-terminated strings, even if they appear to be null-terminated for a particular chip revision.
String fragment order	Strings returned by CPUID functions are broken up into fragments in order to present the data in the CPUID output registers EAX, EBX, ECX, EDX. Thus, you must know the intended order of these fragments in order to reassemble them into the desired string. Transmeta uses two distinct orderings, as shown in the table below: one for strings that fit into three registers, and one for longer strings.

Table 1: String Fragment Order for Strings Returned by CPUID Functions

Function code	String fragment order	String description
0000 0000h	EBX, EDX, ECX	Standard vendor-ID string.
0000 0003h	EAX, EBX, ECX, EDX	Processor serial number (in upper-case hexadecimal)
8000 0000h	EBX, EDX, ECX	Extended vendor-ID string.
8000 0002h	EAX, EBX, ECX, EDX	Preferred processor-name string.
8000 0003h	EAX, EBX, ECX, EDX	
8000 0004h	EAX, EBX, ECX, EDX	
8086 0000h	EBX, EDX, ECX	Transmeta vendor-ID string.
8086 0003h	EAX, EBX, ECX, EDX	Transmeta information string.
8086 0004h	EAX, EBX, ECX, EDX	
8086 0005h	EAX, EBX, ECX, EDX	
8086 0006h	EAX, EBX, ECX, EDX	

1.1.2 Reserved Functions

All CPUID function values other than those listed explicitly in this document are reserved for future extensions. Implicitly reserved functions have zero return values in all output registers.

1.1.3 CPUID Standard Functions

Table 2: CPUID Standard Functions

EAX Input Value	Output Register	Output Value	Output Definition
0000 0000h	EAX	0000 0001h or 0000 0003h	Standard Function Maximum Level Maximum level supported for the standard function. If the processor serial number (PSN) function is disabled, the maximum level supported is 1. If PSN is enabled, the maximum level supported is 3.
	EBX	756E 6547h	Vendor ID String The sequence EBX-EDX-ECX forms the vendor identification string. The value after reset is "GenuineTMx86". This value is programmable using the CPUID_VND1, CPUID_VND2, and CPUID_VND3 MSRs (see <i>CPUID Vendor ID String Registers</i> on page 67 in Chapter 3, <i>Model-Specific Registers</i>). Also see <i>String Representation</i> on page 9.
	ECX	3638 784Dh	
	EDX	5465 6E69h	
0000 0001h	EAX	0000 0543h	Processor Identification Bits are defined as follows: 31-14: Reserved 13-12: CPU type = 0h 11-8: Family = 5 7-4: Model = 4 3-0: Stepping = 3 All bits are programmable using the CPUID_TFMS MSR (see <i>CPUID Type, Family, Model, and Stepping Register</i> on page 66).
	EBX	0000 0000h	Brand Identification Bits 7-0 indicate the brand ID. This function is not supported; this register always returns 0000 0000h. All other bits are reserved.
	ECX	0000 0000h	Reserved.
	EDX	0080 813Fh	Processor Feature Flags All bits not listed are reserved. A set bit indicates the presence of a feature. All bits may be forced to 0 using the CPUID_MASK MSR (see <i>CPUID Feature Flag Mask Registers</i> on page 67 in Chapter 3, <i>Model-Specific Registers</i>). Bits are defined as follows (all bits not listed are reserved): 23: MMX MMX instructions supported 18: PSN Processor serial number 15: CMOV CMOV supported 11: SEP Fast System Call Extension 8: CX8 CMPXCHG8B supported 5: MSR Model specific regs supported 4: TSC Time stamp counter supported 3: PSE Page size extensions supported 2: DE Debug extensions supported 1: VME Virtual mode extensions supported 0: FPU FPU instructions supported

Table 2: CPUID Standard Functions (Continued)

EAX Input Value	Output Register	Output Value	Output Definition
0000 0002h	EAX	0000 0000h	Reserved.
	EBX	0000 0000h	
	ECX	0000 0000h	
	EDX	0000 0000h	
0000 0003h	EAX	0000 0000h	<p>Processor Serial Number (PSN) This function returns the processor serial number. If PSN is disabled, the returned register values are undefined. Due to technical reasons, Transmeta does not guarantee that the PSN is truly unique, and ECX and EDX may return zero values on evaluation parts.</p> <p>OEMs may permanently disable PSN.</p> <p>Also see <i>String Representation</i> on page 9.</p>
	EBX	xxxx xxxh	
	ECX	xxxx xxxh	
	EDX	xxxx xxxh	

1.1.4 CPUID Extended Functions

Table 3: CPUID Extended Functions

EAX Input Value	Output Register	Output Value	Output Definition
8000 0000h	EAX	8000 0006h	Extended Function Maximum Level Maximum level supported for the extended function is 6.
	EBX	6E61 7254h	Vendor ID String The sequence EBX-EDX-ECX forms the vendor identification string. The value after reset is "TransmetaCPU". This value is not programmable. Also see <i>String Representation</i> on page 9.
	ECX	5550 4361h	
	EDX	7465 6D73h	
8000 0001h	EAX	0000 0543h	Processor Identification Bits are defined as follows: 31-14: Reserved 3-12: CPU type = 0 11-8: Family = 5 7-4: Model = 4 3-0: Stepping = 3 This value is not programmable.
	EBX	0000 0000h	Reserved.
	ECX	0000 0000h	Reserved.
	EDX	0081 893Fh	Processor Feature Flags Bits are defined as follows (all bits not listed are reserved): 23: MMX MMX instructions supported 16: FCMOV FCMOV supported 15: CMOV CMOV supported 8: CX8 CMPXCHG8B supported 5: MSR Model specific regs supported 4: TSC Time stamp counter supported 3: PSE Page size extensions supported 2: DE Debug extensions supported 1: VME Virtual mode extensions supported 0: FPU FPU instructions supported This value is not programmable.
8000 0002h	EAX	xxxx xxxxh	Processor Name The output of these function codes, when consecutively stored into memory in the order EAX-EBX-ECX-EDX, forms the processor name. This name may be up to 48 ASCII characters. Unused characters are filled with a value of 00h. This value is not programmable. The value is: Transmeta(tm) Crusoe(tm) Processor TMxxxx where xxxxx stands for the model number, which can be up to 11 characters in length and is derived from an entry in the OEM configuration table. Transmeta highly recommends using this string whenever the processor name is displayed to the end user. Also see <i>String Representation</i> on page 9.
	EBX	xxxx xxxxh	
	ECX	xxxx xxxxh	
	EDX	xxxx xxxxh	
8000 0003h	EAX	xxxx xxxxh	Transmeta(tm) Crusoe(tm) Processor TMxxxx where xxxxx stands for the model number, which can be up to 11 characters in length and is derived from an entry in the OEM configuration table. Transmeta highly recommends using this string whenever the processor name is displayed to the end user. Also see <i>String Representation</i> on page 9.
	EBX	xxxx xxxxh	
	ECX	xxxx xxxxh	
	EDX	xxxx xxxxh	
8000 0004h	EAX	xxxx xxxxh	Transmeta(tm) Crusoe(tm) Processor TMxxxx where xxxxx stands for the model number, which can be up to 11 characters in length and is derived from an entry in the OEM configuration table. Transmeta highly recommends using this string whenever the processor name is displayed to the end user. Also see <i>String Representation</i> on page 9.
	EBX	xxxx xxxxh	
	ECX	xxxx xxxxh	
	EDX	xxxx xxxxh	

Table 3: CPUID Extended Functions (Continued)

EAX Input Value	Output Register	Output Value	Output Definition
8000 0005h	EAX	0000 0000h	Reserved.
	EBX	04FF 04FFh	TLB Characteristics Bits are defined as follows: 31-24: Data TLB associativity = 4-way 23-16: Data TLB size = 256 entries 15-8: Code TLB associativity = 4-way 7-0: Code TLB size = 256 entries
	ECX	4010 0120h	L1 Data Cache Characteristics Bits are defined as follows: 31-24: Cache size = 64 KBytes 23-16: Cache associativity = 16-way 15-8: Cache lines/tag = 1 7-0: Cache bytes/line = 32
	EDX	4008 0140h	L1 Code Cache Characteristics Bits are defined as follows: 31-24: Cache size = 64 KBytes 23-16: Cache associativity = 8-way 15-8: Cache lines/tag = 1 7-0: Cache bytes/line = 64
8000 0006h	EAX	0000 0000h	Reserved.
	EBX	0000 0000h	Reserved.
	ECX	TM5700: 0100 4180h TM5900: 0200 4180h	L2 Cache Characteristics Bits are defined as follows: 31-16: Cache size = 256 KBytes (TM5500) or 512 KBytes (TM5800) 15-12: Cache associativity = 4-way 11-8: Cache lines/tag = 1 7-0: Cache bytes/line = 128
	EDX	0000 0000h	Reserved.

1.1.5 CPUID Transmeta-Specific Functions

Table 4: CPUID Transmeta-Specific Extended Functions

EAX Input Value	Output Register	Output Value	Output Definition
8086 0000h	EAX	8086 0007h	Transmeta-Specific Extended Function Maximum Level Maximum level supported for the extended function is 7.
	EBX	6E61 7254h	Vendor ID String The sequence EBX-EDX-ECX forms the vendor identification string. The value after reset is "TransmetaCPU". This value is not programmable. Also see <i>String Representation</i> on page 9.
	ECX	5550 4361h	
	EDX	7465 6D73h	
8086 0001h	EAX	0000 0543h	Processor Identification Bits are defined as follows: 31-14: Reserved 13-12: CPU type = 0 11-8: Family = 5 7-4: Model = 4 3-0: Stepping = 3 This value is not programmable.
	EBX	<i>aabb ccddh</i> (see below, description)	Crusoe Processor Revision ID The processor version and processor-mask revision. Bits are defined as follows: 31-24: Major processor version 23-16: Minor processor version 15-8: Major processor-mask revision 7-0: Minor processor-mask revision Display the processor revision ID and frequency as follows: <i>a.b-c.d-x</i> <i>a</i> , <i>b</i> , <i>c</i> , and <i>d</i> are the unsigned decimal representations of the numbers <i>aah</i> , <i>bh</i> , <i>cc</i> , and <i>ddh</i> (respectively) from EBX. <i>x</i> is the frequency from ECX, also an unsigned decimal. For example, if EBX is 070B_1311h (7, 11, 19, 17 decimal) and ECX is 1E61h (7777 decimal), display the revision ID as follows: 7.11-19.17-7777
		0200 0000h	See CPUID function 8086 0002h, register EAX
	ECX	xxxx xxxh	Processor Core Frequency This value represents the current processor core clock frequency in MHz. Fractional frequencies may be rounded up or down. See the description for EBX above for printing specifications.
	EDX	xxxx xxxh	Transmeta-Specific Processor Feature Flags A set bit represents the presence of a feature, while a cleared bit indicates either the absence of a feature or a reserved flag. 8: Persistent Translation™ Technology 2.0 7: Persistent Translation™ Technology 1.x 3: LRTI (LongRun Table Interface) 1: LongRun™ support ¹ 0: Recovery active. If this bit is set, the processor is running in a failure recovery mode. This is an indication that the primary version of the Code Morphing Software in the ROM is damaged and should be replaced via upgrade mechanism.

Table 4: CPUID Transmeta-Specific Extended Functions (Continued)

EAX Input Value	Output Register	Output Value	Output Definition
8086 0002h	EAX	xxxx xxxh	If CPUID function 8086 0001h, register EBX reports 0200 0000h, then this register reports the Transmeta processor revision ID in hexadecimal, upper-case.
	EBX	aabb ccddh	Code Morphing Software Revision The first part (the EBX register) consists of four 8-bit-wide numbers: for example, 0102 0304h represents the numbers 1, 2, 3, and 4. The second part (the ECX register) consists of one 32-bit-wide number. Display the Code Morphing Software revision ID as follows: <i>a.b.c-d-x...</i> <i>a, b, c, d, and x</i> are the unsigned decimal representations of the numbers <i>aah, bbh, cch, ddh, and xxxh xxxh</i> from registers EBX and ECX, respectively. For example, if EBX is 0509 011Fh (5, 9, 1, 31 decimal) and ECX is 0041 A028h (4300840 decimal), display the revision ID as follows: 5.9.1-31-4300840
	ECX	xxxx xxxh	
	EDX	0000 0000h	Reserved.
8086 0003h	EAX	xxxx xxxh	Transmeta-Specific Information String The output of these function codes, when consecutively stored into memory in the order EAX-EBX-ECX-EDX, forms a Transmeta-specific information string which may consist of up to 64 ASCII characters. Unused characters are filled with a value of 00h. Also see <i>String Representation</i> on page 9.
	EBX	xxxx xxxh	
	ECX	xxxx xxxh	
	EDX	xxxx xxxh	
8086 0004h	EAX	xxxx xxxh	
	EBX	xxxx xxxh	
	ECX	xxxx xxxh	
	EDX	xxxx xxxh	
8086 0005h	EAX	xxxx xxxh	
	EBX	xxxx xxxh	
	ECX	xxxx xxxh	
	EDX	xxxx xxxh	
8086 0006h	EAX	xxxx xxxh	
	EBX	xxxx xxxh	
	ECX	xxxx xxxh	
	EDX	xxxx xxxh	
8086 0007h	EAX	xxxx xxxh	Current Processor Core Frequency This value represents the current processor core clock frequency in MHz. Fractional frequencies may be rounded up or down. This value is 0000 0000h if LongRun is not supported.
	EBX	xxxx xxxh	Current Processor Core Voltage This value represents the current processor core voltage, in millivolts. This value is 0000 0000h if LongRun is not supported.
	ECX	xxxx xxxh	Current Performance Percentage This value represents the current performance percentage (0-100d). This value is 0000 0000h if LongRun is not supported.
	EDX	0000 0000h	Reserved.

1. Disabled LongRun™. The feature flag remains set even when LongRun has been effectively disabled by programming the LONGRUN MSR to a performance percentage of 100%:100% (see *LongRun™ Control and Status Register* on page 68).

1.2 Frequency Detection

The recommended method for determining the frequency of the TM5700/TM5900 is to use the Transmeta-specific extended functions of the CPUID instruction, as follows:

- Executing CPUID with an input value of 8086 0001h always returns the nominal (i.e. maximum) CPU core frequency in the ECX register.
- If LongRun is enabled, the actual CPU frequency can be determined from CPUID Function 8086 0007h, EAX register. See *Processor Core Frequency* on page 15.

You can determine whether LongRun is enabled with CPUID Function 8086 0001, EDX register, bit 1 (see *Transmeta-Specific Processor Feature Flags* on page 15).

Memory and PCI bus frequencies are available in the PCI configuration registers described in Chapter 2, *PCI Configuration Registers*.

PCI Configuration Registers

Note

Any future upgrade to CMS for the TM5700/TM5900 will be compatible with older OEM configuration tables and BIOS configurations. In particular, no VNB registers will change in name or function.

The Crusoe™ Processor Model TM5700/TM5900 contains two sets of configuration registers accessible by software via the host CPU I/O address space: I/O-mapped registers and PCI configuration registers. I/O mapped registers provide access to the PCI configuration registers. Both sets of registers are accessible only by the host processor—they cannot be accessed by PCI bus masters.

The registers can be accessed as byte, word or dword quantities unless otherwise specified. As required by the PCI Revision 2.1 specification, all multiple-byte numeric fields use *little-endian* ordering (the least significant bit corresponds to the lowest address). Registers are classified as either read-only, read/write, read/write clear, read-after-write-once, or reserved, as defined in the following table:

Table 5: PCI Register Access Classifications

Access Classification	Symbol	Read Effect	Write Effect
Read-only	RO	Reads contents of register.	None.
Read/write	R/W	Reads contents of register.	Writes data to register.
Read/write clear	R/WC	Reads contents of register.	Write of 1 sets the corresponding bit to zero. Write of 0 has no effect.
Read after write-once	R/WO	Reads contents of register.	Write of non-zero value can only be changed by CPU RESET (i.e. not init).
Reserved	--	Always returns 0.	None.

The TM5700/TM5900 contains address locations in the PCI configuration space that are *reserved*. The TM5700/TM5900 responds to these addresses by completing the cycle and returning a value of 0. Software should not write to reserved configuration registers in the device-specific address region (above offset 3Fh).

Some of the PCI configuration registers listed contain *reserved bits*. Software should not modify or rely on the value of reserved bits.

2.1 SMRAM Differences

The Transmeta Virtual Northbridge (VNB) embedded virtual host bridge is designed to achieve maximal compatibility with the Intel 82443BX Northbridge.

The Intel 82443BX SMRAM programming model is not 100% compatible with the Transmeta SMRAM programming model. The differences stem from the fact that the Intel 82443BX Northbridge supports split code/data accesses while the CPU executes in SMM mode, whereas the Transmeta VNB does not. As a result, the Transmeta northbridge has to relocate the video frame buffer out of the Compatibility SMRAM region (located in A0000h to BFFFFh) into a user-specified virtual video window.

The following table summarizes the resulting SMRAM differences between a Intel 82443BX Northbridge and the Transmeta Virtual Northbridge.

Table 6: SMRAM Differences Between 82443BX and Transmeta VNB

	Intel 82443BX Northbridge		Transmeta Virtual Northbridge	
	Transaction	DRAM	Transaction	DRAM
Compatible A region	A0000h to BFFFFh <ul style="list-style-type: none"> not write-back cacheable optional split code/data accesses 	A0000h to BFFFFh	A0000h to BFFFFh <ul style="list-style-type: none"> write-back cacheable no split code/data accesses 	TOM+896K to TOM+1024K <ul style="list-style-type: none"> DMA accesses must be directed to physical DRAM rather than SMM Transaction space
High H region	100A0000h to 100BFFFFh <ul style="list-style-type: none"> write-back cacheable 	A0000h to BFFFFh	TOM+896K to TOM+1024K <ul style="list-style-type: none"> DMA accesses must be directed to physical DRAM rather than SMM Transaction space 	
Extended T region	TOM-TSEG_SZ+256M to TOM+256M <ul style="list-style-type: none"> write-back cacheable 	TOM-TSEG_SZ to TOM	TOM to TOM+896K <ul style="list-style-type: none"> write-back cacheable 	

2.2 I/O Mapped Registers

The TM5700/TM5900 Virtual Northbridge provides two I/O-mapped registers that in turn provide access to all other PCI configuration registers: the Configuration Address Register (CONFADDR) and the Configuration Data Register (CONFDATA). A third I/O mapped register (PM_CR2) is used to disable PCI bus master activity prior to entering a CPU stop clock state.

The I/O addresses and bit definitions for these registers are listed in the following sections.

Table 7: I/O Mapped Register Summary

I/O Location	Register Description	Register Name	Size	Access	Default Value	Page
0022h ¹	ACPI Power Management Control Register	PM_CR2	8 Bits	R/W	00h	21
0CF8h	Configuration Address Register	CONFADDR	32 Bits	R/W (DWORD only)	0000 0000h	22
0CFCh	Configuration Data Register	CONFDATA	32 Bits	R/W	0000 0000h	23

1. This value is programmable via the ACPI Power Management Control Address Register (PM_CR2_ADDR).

ACPI Power Management Control Register

I/O Location	Register Name	Size	Access	Default Value
0022h ¹	PM_CR2	8 Bits	R/W	00h

- This value is programmable via the ACPI Power Management Control Address Register (PM_CR2_ADDR).

The PM_CR2 register is used to disable the PCI arbiter to prevent any external bus masters from acquiring the PCI bus. Any currently running PCI cycles terminate properly.

Accesses to this register are controlled by the power management control registers:

- PM_CR2_ADDR (offset 78-79h): This register specifies the address where the PM_CR2 register is mapped into I/O space. It defaults to 22h, which is compatible to the Intel 82443BX Northbridge and PIIX4 Southbridge.
- PM_CR.PM_CR2_EN (offset 7Ah, bit 6):
 - 0: I/O accesses to location PM_CR2 are forwarded to the PCI bus.
 - 1: I/O accesses to location PM_CR2 go to the ACPI Register. ?
- PM_CR2_OPT.PM_CR2_FORWARD (offset 7Bh, bit 6): The Transmeta VNB supports forwarding of all PM_CR2 register accesses to the PCI bus, which allows other system components, such as the Southbridge, to snoop them:
 - 0: I/O accesses to PM_CR2 are handled internally.
 - 1: I/O accesses to PM_CR2 also appear on the PCI bus.

Note

In order to boot from a parallel ROM, the OEM configuration table needs to include some init cycles to program the southbridge to open the aperture window into the ROM to at least 512K (as opposed to the 64K–128K aperture available at reset). This is typically done by issuing PCI configuration cycles that set some registers in the southbridge.

The BIOS designer must be aware of this issue and not depend on the state of these southbridge registers persisting across a suspend-to-RAM (S3 suspend state) transition. The low-level resume from STR code in the TM5700/TM5900 boot code re-issues these cycles on the way out of STR (S3 resume), so these registers (and the aperture into ROM) may appear to change to the BIOS. If the BIOS depends on their setting across STR, it can merely save and restore them.

The following table illustrates the setting of PM_CR2_ADDR and PM_CR2_OPT.PM_CR2_FORWARD for two well-known Southbridges:

Southbridge	PM_CR2_ADDR	PM_CR2_OPT.PM_CR2_FORWARD
Intel PIIX4	0x0022	0
ALi M1535	0x0030	1

The following table illustrates the effect of PM_CR.PM_CR2_EN and PM_CR2_OPT.PM_CR2_FORWARD on accesses to PM_CR2:

PM_CR.PM_CR2_EN	PM_CR2_OPT.PM_CR2_FORWARD	PM_CR2 access at PM_CR2_ADDR	PCI bus cycle at PM_CR2_ADDR
0	0	no	yes
0	1	no	yes
1	0	yes	no
1	1	yes	yes

Bit No.	Bit Name	Function Description
7:1	--	Reserved.
0	ARB_DIS	<p>Arbiter Disable When set (1), the processor's PCI controller ignores requests from all other PCI bus masters. This feature should be used prior to placing the CPU in a stop clock state to guarantee cache coherency while the CPU clock is stopped. If the PCI-to-ISA bridge is in passive release mode, masking of the P_HOLD# request line does not occur until an active release is seen via assertion of P_HLDA#, preventing any possible deadlock.</p> <p>The PM_CR2_EN bit in the PM_CR register must be set prior to accessing this register.</p>

Configuration Address Register

I/O Location	Register Name	Size	Access	Default Value
0CF8h	CONFADDR	32 Bits	R/W (DWORD only)	0000 0000h

CONFADDR is a 32 bit register accessed only when referenced as a DWord. A Byte or Word reference will "pass through" the Configuration Address Register onto the PCI bus as an I/O cycle. The CONFADDR register contains the Bus Number, Device Number, Function Number, and Register Number for which a subsequent configuration access is intended.

CONFADDR Bit Definitions

Bit No.	Bit Name	Function Description
31	CFGE	<p>Configuration Enable If set (1), PCI configuration space access is enabled. If clear (0), configuration space access is disabled.</p>
30:24	--	Reserved.
23:16	BUS	<p>Bus Number When the bus number is 00h, the target of the configuration cycle is the TM5700/TM5900 or the PCI local bus connected directly to the TM5700/TM5900. If the bus number is 00h and the target is not the TM5700/TM5900 itself, the TM5700/TM5900 generates a type 0 configuration cycle on the PCI bus. If the bus number is non-zero, the TM5700/TM5900 generates a type 1 configuration cycle and the bus number is mapped to AD(23:16) during the address phase of the PCI cycle.</p>

CONFADDR Bit Definitions (Continued)

Bit No.	Bit Name	Function Description
15:11	DEV	Device Number This field specifies a specific device on the PCI bus. For a type 0 configuration cycle, this field is decoded and only one bit of AD(31:11) is driven to a 1 during the address phase of the PCI cycle. The TM5700/TM5900 is always device number 0 and does not pass its configuration cycles to the PCI bus. Therefore, AD ₁₁ is never asserted during a type 0 configuration cycle. The highest device number allowed is 20 which corresponds to bit 31. All device numbers higher than 20 cause a type 0 configuration cycle on the PCI bus with no IDSEL asserted. For a type 1 configuration cycle, this field is mapped directly to AD(15:11) during the address phase of the PCI cycle.
10:8	FUNC	Function Number This field specifies a specific function within the device specified by the bus number and device number. This field is mapped directly to AD(10:8) during the address phase of the PCI cycle. The TM5700/TM5900 only responds to configuration cycles with a function number of 000b. Configuration cycles attempted with bus number 0, device number 0, and a non-zero function number generate a type 0 configuration cycle on the PCI bus with no IDSEL asserted.
7:2	REG	Register Number This field specifies a specific register within the function specified by the bus number, device number and function number. This field is mapped directly to AD(7:2) during the address phase of the PCI cycle.
1:0	--	Reserved.

Configuration Data Register

I/O Location	Register Name	Size	Access	Default Value
0CFCh	CONFDATA	32 Bits	R/W	0000 0000h

CONFDATA is a 32 bit read/write window into configuration space. The portion of configuration space that is referenced by CONFDATA is determined by the contents of CONFADDR.

CONFDATA Bit Definitions

Bit No.	Bit Name	Function Description
31:0	CDW	Configuration Data Window 32-bit read/write window into configuration space. The address of the configuration register that is accessed is determined by the contents of the CONFADDR register. The Configuration Enable bit (CFGE, bit 31) of CONFADDR must be set (1) to enable a configuration space access.

2.3 PCI Configuration Registers

The TM5700/TM5900 VNB contains a block of PCI configuration registers that provide access to the TM5700/TM5900's configuration space header (addresses 00h to 3Fh) and to TM5700/TM5900 specific configuration functions (above 3Fh). The following table lists the PCI configuration address assignments for these registers. The register and bit definitions for each of the registers are listed in the following sections.

Three functions of PCI configuration registers are available to configure:

- Host-to-PCI Bridge Registers (Function 0)
- SDRAM Memory Controller Registers (Function 1)
- Extended BIOS Scratch Pad (Function 2)

2.3.1 Host-to-PCI Bridge Registers (Function 0)

Host-to-PCI Bridge Registers are summarized in the following table in order by PCI address space offset. Detailed explanations of these registers are shown in the sections following the table.

Table 8: PCI Registers (Function 0) Summary

Address Offset	Register Description	Register Name	Size	Access	Default Value	Page
00-01h	Vendor Identification Register	VID0	16 Bits	RO	1279h	27
02-03h	Device Identification Register	DID0	16 Bits	RO	0395h	27
04-05h	PCI Command Register	PCICMD	16 Bits	R/W	0006h	27
06-07h	PCI Status Register	PCISTS	16 Bits	RO and R/WC	0000h	28
08h	Revision Identification Register	RID0	8 Bits	RO	4h	29
09h	Programming Interface Register	PI0	8 Bits	RO	00h	29
0Ah	Sub-Class Code Register	SUBC0	8 Bits	RO	00h	29
0Bh	Base Class Code Register	BCC0	8 Bits	RO	06h	30
0C	Reserved	--		--	--	
0Dh	Master Latency Timer	MLT	8 Bits	R/W	See description	30
0Eh	Header Type Register	HEDT0	8 Bits	RO	00h	30
0Fh	Reserved	--		--	--	
10-13h	Virtual (Video) Window Base Address Register	VWBASE	32 Bits	R/W	0000 0000h	31
14-2Bh	Reserved	--		--	--	
2C-2Dh	Subsystem Vendor Identification Register	SVID	16 Bits	R/WO	See description (also see note ¹)	32
2E-2Fh	Subsystem Identification Register	SID	16 Bits	R/WO	See description (also see note ¹)	32
30-47h	Reserved	--		--	--	
4A-4Bh	Top of Memory Register	TOM	16 Bits	RO	Memory capacity	32
4Ch	Size of Code Morphing Software Memory Region Register	CMSMEM	16 Bits	R/W	Memory capacity	33
4Eh	Maximum Size of Code Morphing Software Memory Region Register	CMSMAX	16 Bits	RO	Memory capacity	33
4F-58h	Reserved	--		--	--	

Table 8: PCI Registers (Function 0) Summary (Continued)

Address Offset	Register Description	Register Name	Size	Access	Default Value	Page
59h	Programmable Memory Attribute Registers	PAB0	8 Bits	R/W	0Fh	34
5Ah		PAB1			00h	
5Bh		PAB2				
5Ch		PAB3				
5Dh		PAB4				
5Eh		PAB5				
5Fh		PAB6				
60-71h	Reserved	--		--	--	
72h	System Management RAM Control Register	SM_RAM_CR	8 Bits	R/W	02h	35
73h	Extended SMM RAM Control Register	ESM_RAM_CR	8 Bits	R/W	3Fh	36
78-79h	ACPI Power Management Control Address Register	PM_CR2_ADDR	16 Bits	R/W	0022h	37
7Ah	Power Management Control Register	PM_CR	8 Bits	R/WO	18h	37
7Bh	ACPI Power Management Control Options Register	PM_CR2_OPT	8 Bits	R/W	00h	38
7C-7Eh	Reserved	--		--	--	
7Fh	Power Management Control Register	PM_CONTROL	8 Bits	R/W	02h (see description)	38
80-9Fh	Reserved	--		--	--	
A0h	Configuration Locks Register	LOCK	8 Bits	R/WO	01h	39
A1h	ROM Control Register	ROMCTL	8 Bits	R/W*	06h (see description)	39
A1-A3h	Reserved	--		--	--	
A4-A7h	OEM-Defined Meaning Register	OEMOPT	32 Bits	--	See description	41
A8h	LongRun Advanced Thermal Management Definitions Register	LR_ATM	8 Bits	R/W	01h	41
A9h	Processor Performance Control	PERF_CTRL	8 Bits	R/W	See description	42
AA-ABh	Reserved	--		--	--	
AC-AFh	PCI Arbiter Control	PCI_ARB_CTRL	32 bits	R/W	See description	43
B0-CFh	Reserved	--		--	--	
D0-D7	BIOS Scratch Pad Register	BSPAD	64 Bits	R/W	0000 0000 0000 0000h	43
D8-D9h	Power Management Power-On Suspend Address Register	POS_ADDR	16 Bits	R/W	0000h	44
DA-DBh	Power Management Power-On Suspend Data Mask Register	POS_MASK	16 Bits	R/W	0000h	44
DC-DDh	Power Management Power-On Suspend Data Register	POS_DATA	16 Bits	R/W	0000h	44
DEh	Power Management Power-On Suspend Control Register	POS_CTL	8 Bits	R/W	00h	45
DFh	Reserved	--		--	--	

Table 8: PCI Registers (Function 0) Summary (Continued)

Address Offset	Register Description	Register Name	Size	Access	Default Value	Page
E0-E1h	Power Management Level 2 Address Register	P_LVL2_ADDR	16 Bits	R/W	0000h	45
E2-E3h	Power Management Level 2 Data Mask Register	P_LVL2_MASK	16 Bits	R/W	0000h	45
E4-E5h	Power Management Level 2 Compare Data Register	P_LVL2_DATA	16 Bits	R/W	0000h	46
E6h	Power Management Level 2 Control Register	P_LVL2_CTL	8 Bits	R/W	00h	46
E7h	Reserved	--		--	--	
E8-E9h	Power Management Level 3 Address Register	P_LVL3_ADDR	16 Bits	R/W	0000h	46
EA-EBh	Power Management Level 3 Data Mask Register	P_LVL3_MASK	16 Bits	R/W	0000h	47
EC-EDh	Power Management Level 3 Compare Data Register	P_LVL3_DATA	16 Bits	R/W	0000h	47
EEh	Power Management Level 3 Control Register	P_LVL3_CTL	8 Bits	R/W	00h	47
EFh	Reserved	--		--	--	
F0-F1h	Power Management Suspend to RAM Address Register	STR_ADDR	16 Bits	R/W	0000h	48
F2-F3h	Power Management Suspend to RAM Data Mask Register	STR_MASK	16 Bits	R/W	0000h	48
F4-F5h	Power Management Suspend to RAM Compare Data Register	STR_DATA	16 Bits	R/W	0000h	48
F6h	Power Management Suspend to RAM Control Register	STR_CTL	8 Bits	R/W	00h	49
F7-FBh	Reserved	--		--	--	
FC-FDh	Master I/O Clock Base Frequency Register	MASTER_CLK	16 Bits	RO	0000h (See note ²)	49
FEh	Memory Clock Divisors Register	MEMDIV	8 Bits	RO	00h (See note ²)	49
FFh	PCI Clock Divisor Register	PCIDIV	8 Bits	RO	00h (See note ²)	50

1. This field should either be preset by the OEM Configuration Table (recommended), or initialized by the BIOS during the boot process. If preset by the OEM Configuration Table, this field is read only. If not initialized by the OEM Configuration Table, this field has a default value of zero and is programmable until a non-zero value is written to it.
2. Reset values for these registers vary depending on system configuration. Registers that are designated as RO (read-only) do **not** have to be initialized by the BIOS.

00-01h Vendor Identification Register

Addr Offset	Register Name	Size	Access	Default Value
00-01h	VID0	16 Bits	RO	1279h

The VID0 register contains the vendor identification number. This register, in combination with the DID0 register, uniquely identifies the TM5700/TM5900. Writes to this register have no effect.

VID0 Bit Definitions

Bit No.	Name	Function Description
15:0	VID	Vendor Identification Number This 16-bit value is the PCI vendor identification number assigned to Transmeta. Transmeta VID = 1279h.

02-03h Device Identification Register

Addr Offset	Register Name	Size	Access	Default Value
02-03h	DID0	16 Bits	RO	0395h

The DID0 register contains the device identification number. This 16-bit register, in combination with the VID0 register, uniquely identifies the TM5700/TM5900. Writes to this register have no effect.

DID0 Bit Definitions

Bit No.	Name	Function Description
15:0	DID	Device Identification Number This 16-bit value is the identification number assigned to the TM5700/TM5900 power management controller (). Transmeta DID = 0395h

04-05h PCI Command Register

Addr Offset	Register Name	Size	Access	Default Value
04-05h	PCICMD	16 Bits	R/W	0006h

The PCICMD register provides basic control of the TM5700/TM5900 PCI interface response to PCI cycles. This register enables/disables the SERR# signal, configures the TM5700/TM5900 response to PCI special cycles, and enables/disables PCI bus master access to main memory.

PCICMD Bit Definitions

Bit No.	Name	Function Description
15:10	--	Reserved.
9	FB2BC	Fast Back-to-Back This bit is always 0.
8	SERRE	SERR# Enable This bit is always 0.
7	ADSTEP	Address/Data Stepping This bit is always 0.
6	PERRE	Parity Error Enable This bit is always 0.
5	VPS	Video Palette Snooping This bit is always 0.
4	MWIE	Memory Write and Invalidate Enable This bit is always 0.

PCICMD Bit Definitions (Continued)

Bit No.	Name	Function Description
3	SCE	Special Cycle Enable This bit is always 0.
2	BME	Bus Master Enable This bit is always 1. The TM5700/TM5900 does not support disabling of its PCI bus master capability.
1	MAE	Memory Access Enable This bit enables or disables PCI masters from accessing memory: 1: The TM5700/TM5900 permits PCI masters to access main memory if the PCI address selects enabled DRAM space. 0: The TM5700/TM5900 prohibits PCI masters from accessing main memory.
0	IOAE	I/O Access Enable This bit is always 0. The TM5700/TM5900 does not respond to PCI I/O cycles.

06-07h PCI Status Register

Addr Offset	Register Name	Size	Access	Default Value
06-07h	PCISTS	16 Bits	RO and R/WC	0000h

The PCISTS register reports PCI master abort and PCI target abort status on the PCI bus. PCISTS also indicates the DEVSEL# timing set by the TM5700/TM5900 PCI interface for target responses on the PCI bus.

PCISTS Bit Definitions

Bit No.	Name	Function Description
15	DPE	Detected Parity Error This bit is always 0. The TM5700/TM5900 does not support PCI received parity checking. Bit is read only.
14	SSE	Signaled System Error This bit is always 0. The TM5700/TM5900 does not support SERR#. Bit is read only.
13	RMAS	Received Master Abort Status This bit is set (1) when the TM5700/TM5900 terminates a host-to-PCI transaction with an unexpected master abort. Software resets this bit to 0 by writing a 1 to it.
12	RTAS	Received Target Abort Status This bit is set (1) when a TM5700/TM5900-initiated PCI cycle is terminated with a target abort. Software resets this bit to 0 by writing a 1 to it.
11	STAS	Signaled Target Abort Status This bit is always a 0. The TM5700/TM5900 never terminates a PCI cycle with a target abort. Bit is read only.
10:9	DEVT	DEVSEL# Timing This field specifies the timing of the DEVSEL# signal when the TM5700/TM5900 responds as a PCI target. This field is always 01b, which indicates medium timing. Bits are read only.
8	DPD	Data Parity Detected This bit is always 0. The TM5700/TM5900 does not support PERR#. Bit is read only.
7	FB2BS	Fast Back-to-Back This bit is always 0. The TM5700/TM5900 does not support fast back-to-back cycle generation. Bit is read only.
6	UDF	User Defined Format This bit is always 0. The TM5700/TM5900 does not contain any configurations that depend on the environment. Bit is read only.
5	66C	66 MHz PCI Capable This bit is always 0. The TM5700/TM5900 does not support 66 MHz PCI. Bit is read only.
4:0	--	Reserved. Bits are read only.

08h Revision Identification Register

Addr Offset	Register Name	Size	Access	Default Value
08h	RID0	8 Bits	RO	4h

The RID0 register contains the TM5700/TM5900 VNB function 0 revision identification number. Writes to this register have no effect.

RID0 Bit Definitions

Bit No.	Name	Function Description
7:0	RID	Revision Identification Number This 8-bit value is the revision identification number assigned to the TM5700/TM5900 power management controller.

09h Programming Interface Register

Addr Offset	Register Name	Size	Access	Default Value
09h	PI0	8 Bits	RO	00h

This field is read-only.

PI0 Bit Definitions

Bit No.	Name	Function Description
7:0	PGIF	Programming Interface This read-only field always returns 0 when read, since none of the VNB function 0 subclasses have further division into programming interfaces.

0Ah Sub-Class Code Register

Addr Offset	Register Name	Size	Access	Default Value
0Ah	SUBC0	8 Bits	RO	00h

The SUBC0 register contains the TM5700/TM5900 VNB category code. The default value 00h indicates a host bridge. Writes to this register have no effect.

SUBC0 Bit Definitions

Bit No.	Name	Function Description
7:0	SUBC	Sub-Class Code This 8-bit value indicates the category type for the TM5700/TM5900 PCI bridge. The default value 00h indicates a host bridge.

0Bh Base Class Code Register

Addr Offset	Register Name	Size	Access	Default Value
0Bh	BCC0	8 Bits	RO	06h

The BCC0 register contains the TM5700/TM5900 VNB base class code. The default value 06h indicates a bridge device. Writes to this register have no effect.

BCC0 Bit Definitions

Bit No.	Name	Function Description
7:0	BASEC	Base Class Code This 8-bit value indicates the base class code for the TM5700/TM5900 PCI bridge. The default value 06h indicates a bridge device.

0Dh Master Latency Timer

Addr Offset	Register Name	Size	Access	Default Value
0Dh	MLT	8 Bits	R/W	See description

This register specifies, in units of PCI bus clocks, the value of the Latency Timer for this PCI bus master.

The initial value is determined by OEM config table entry `initial_pci_latency_timer`. See the *Development and Manufacturing Guide* for details.

MLT Bit Definitions

Bit No.	Name	Function Description
7:0	MLT	Master Latency Timer Controls the maximum number of PCI bus cycles the TM5x00 can burst to the PCI bus when acting as a bus master. A value of 0 specifies a maximum of 256 PCI bus cycles.

0Eh Header Type Register

Addr Offset	Register Name	Size	Access	Default Value
0Eh	HEDT0	8 Bits	RO	00h

The HEDT0 register defines the TM5700/TM5900 VNB header type of the configuration space. Writes to this register have no effect.

HEDT0 Bit Definitions

Bit No.	Name	Function Description
7:0	HEADT	Header Type This 8-bit value indicates the header type for the TM5700/TM5900 PCI bridge. The default value 00h is always returned by reads to this register.

10-13h Virtual (Video) Window Base Address Register

Addr Offset	Register Name	Size	Access	Default Value
10-13h	VWBASE	32 Bits	R/W	0000 0000h

The VWBASE register provides a mechanism for handling the legacy video frame buffer within the PCI architecture.

When SMM remapping is enabled (only), the A0000-BFFFFh region will be shadowed at address xxxA0000-xxxBFFFFh. The base address (xxx00000 above) is specified via a standard PCI address range register in the TM5700/TM5900 power management controller (i.e. the TM5700/TM5900 will claim the region).

Standard PCI configuration code will assign an address range to this register; e.g. if VWBASE is set to 8000 0000h, the video frame buffer would be mapped from 800A0000-800BFFFFh.

The video buffer on a legacy video card (A0000-BFFFFh) has unknown prefetchable characteristics. Therefore, the virtual video window is considered non-prefetchable.

The BIOS or OS may change this register at any time. Therefore, the SMM handler must read the VWBASE register to obtain the actual virtual video window base address. The value read from this register may then be used as a base address to read and write the actual video frame buffer at offset A0000-BFFFFh (VWBASE + A0000h to VWBASE + BFFFFh).

VWBASE Bit Definitions

Bit No.	Name	Function Description
31:20	VW_BASE	Virtual Window Base Memory Address These bits define the base address of a buffer that contains the legacy video frame buffer. This feature allows the system management mode (SMM) handler to access the legacy video frame buffer at offset A0000-BFFFFh within this 1 MByte memory buffer (VWBASE + A0000h to VWBASE + BFFFFh). This memory window only exists under the following conditions: <ul style="list-style-type: none"> when SMM memory is enabled and the processor is in SMM when SMM memory space is explicitly made visible (i.e., SM_OPEN is set to 1).
19:4	VW_ALIGN	Virtual Window Memory Area Aligned Address These bits specify bits 19:4 of the virtual video window base address. These bits default to 0 and are read-only to guarantee that PCI configuration software assigns the video base address at a 1 MByte boundary.
3	VW_PREFETCH	Virtual Window Memory Area Prefetchable This bit is read only and set to zero to indicate that the video window memory area is not prefetchable.
2:1	VW_TYPE	Virtual Window Memory Address Type These bits are read only and set to zero to indicate that the video window base address may be located anywhere in 32-bit address space.
0	VW_MEMORY	Virtual Window Memory Device This bit is read only and set to zero to indicate the video memory maps into memory space rather than I/O space.

2C-2Dh Subsystem Vendor Identification Register

Addr Offset	Register Name	Size	Access	Default Value
2C-2Dh	SVID	16 Bits	R/WO	See description

Subsystem Vendor Identification Register, to be set by the OEM Configuration Table or in BIOS. For more information about the OEM configuration table, see the *Development and Manufacturing Guide*.

SVID Bit Definitions

Bit No.	Name	Function Description
15:0	SVID	Subsystem Vendor ID This value should be preset by either the OEM configuration table (recommended), or by the BIOS during boot-up. If preset by the OEM configuration table, the field is read-only. If preset by the BIOS, the field is writable only until a non-zero value is written to it, and is read-only after.

2E-2Fh Subsystem Identification Register

Addr Offset	Register Name	Size	Access	Default Value
2E-2Fh	SID	16 Bits	R/WO	See description

Subsystem Identification Register, to be set by the OEM Configuration Table or in BIOS. For more information about the OEM configuration table, see the *Development and Manufacturing Guide*.

SID Bit Definitions

Bit No.	Name	Function Description
15:0	SID	Subsystem ID This value should be preset by either the OEM configuration table (recommended), or by the BIOS during boot-up. If preset by the OEM configuration table, the field is read only. If preset by the BIOS, the field is writable only until a non-zero value is written to it, and is read-only after that.

4A-4Bh Top of Memory Register

Addr Offset	Register Name	Size	Access	Default Value
4A-4Bh	TOM	16 Bits	RO	Memory capacity

Top of CMS memory.

TOM Bit Definitions

Bit No.	Name	Function Description
15:0	TOM	Top of Memory These bits specify bits 31-16 of the starting address of the memory allocated to Code Morphing Software. The value in this register is aligned to a MByte boundary (bits 3-0 of this register will be zero). PCI configuration software should not assign device addresses below TOM + 1 MB.

4Ch Size of Code Morphing Software Memory Region Register

Addr Offset	Register Name	Size	Access	Default Value
4Ch	CMSMEM	16 Bits	R/W	Memory capacity

This register specifies the amount of memory, in units of 64 KB, allocated to Code Morphing software. Writes to this register of a value greater than the value of the CMSMAX register (see below) or less than the defined minimum (usually 8 MB) will be adjusted to match the largest or smallest possible value, respectively.

The initial value for this field comes from the OEM configuration table field cms_ram_size.

Note: the readonly part of this register is not guaranteed to be zero.

CMSMEM Bit Definitions

Bit No.	Name	Function Description
15:6	CMSMEM	Code Morphing Software Memory Size Writable portion of Code Morphing software memory. Can only be adjusted in multiples of 4 MB.
5:0	CMSMEM	Code Morphing Software Memory Size (Read-Only Portion) Read-only portion of Code Morphing software memory. Software MUST NOT alter these bits.

4Ch Maximum Size of Code Morphing Software Memory Region Register

Addr Offset	Register Name	Size	Access	Default Value
4Eh	CMSMAX	16 Bits	RO	Memory capacity

This register specifies the maximum amount of memory, in units of 64 KB, which can be allocated to Code Morphing Software.

The bits in this register which correspond to the readonly portion of the CMSMEM register will always have the same value as the corresponding bits in that register.

CMSMAX Bit Definitions

Bit No.	Name	Function Description
15:0	CMSMAX	Code Morphing Software Maximum Memory Size

59h-5Fh Programmable Memory Attribute Registers

Addr Offset	Register Name	Register Details				Default Value	Reg Size	Access
		Bits	Address	Size	Range			
59h	PAB0	3:0	Reserved			0Fh ¹	8 Bits	R/W
		7:4	960 KB	64 KB	F0000 - FFFFFh			
5Ah	PAB1	3:0	768 KB	16 KB	C0000 - C3FFFh	00h	8 Bits	R/W
		7:4	784 KB	16 KB	C4000 - C7FFFh			
5Bh	PAB2	3:0	800 KB	16 KB	C8000 - CBFFFh	00h	8 Bits	R/W
		7:4	816 KB	16 KB	CC000 - CFFFh			
5Ch	PAB3	3:0	832 KB	16 KB	D0000 - D3FFFh	00h	8 Bits	R/W
		7:4	848 KB	16 KB	D4000 - D7FFFh			
5Dh	PAB4	3:0	864 KB	16 KB	D8000 - DBFFFh	00h	8 Bits	R/W
		7:4	880 KB	16 KB	DC000 - DFFFFh			
5Eh	PAB5	3:0	896 KB	16 KB	E0000 - E3FFFh	00h	8 Bits	R/W
		7:4	912 KB	16 KB	E4000 - E7FFFh			
5Fh	PAB6	3:0	928 KB	16 KB	E8000 - EBFFFh	00h	8 Bits	R/W
		7:4	944 KB	16 KB	EC000 - EFFFFh			

- For PAB0, the default value is 0Fh, indicating that reads and writes are sent to SDRAM for the 80000 - 9FFFFh memory region. All other PAB_n registers have default value of 00h.

Programmable memory attribute registers.

PAB0–PAB6 Bit Definitions

Bit No.	Name ¹	Function Description
7:6	--	Reserved.
5	WE1 _{nn}	Write Enable Region 1 If set, writes to the specified address range are directed to SDRAM. If clear, writes to the region are directed to the PCI bus.
4	RE1 _{nn}	Read Enable Region 1 If set, reads from the specified address range are directed to SDRAM. If clear, reads from the region are directed to the PCI bus.
3:2	--	Reserved.
1	WE0 _{nn}	Write Enable Region 0 If set, writes to the specified address range are directed to SDRAM. If clear, writes to the region are directed to the PCI bus.
0	RE0 _{nn}	Read Enable Region 0 If set, reads from the specified address range are directed to SDRAM. If clear, reads from the region are directed to the PCI bus.

- nn* = bits 12-19 of the first address of the applicable address region. For example:
PAB1(0) = RE0_C0
PAB2(4) = RE1_CC
See the register name table above for corresponding memory regions.

72h System Management RAM Control Register

Addr Offset	Register Name	Size	Access	Default Value
72h	SM_RAM_CR	8 Bits	R/W	02h

The SM_RAM_CR register controls how accesses to compatibility and extended SMRAM spaces are treated. The SM_OPEN and SM_LOCK bits function only when the GSM_RAM_EN bit is set to 1. The SM_OPEN bit must be reset before the SM_LOCK bit is set.

SM_RAM_CR Bit Definitions

Bit No.	Name	Function Description
7	--	Reserved.
6	SM_OPEN	SMM Space Open When SM_LOCK is set to 0 and SM_OPEN is set to 1, SMM memory space is made visible even when SMM decode is not active. This is intended to help BIOS initialize SMM space, and this bit may be used to initialize SMM memory prior to servicing the first SMI. When SM_LOCK is set to 1, SM_OPEN is reset to 0 and becomes read only.
5	--	Reserved.
4	SM_LOCK	SMM Space Locked When SM_LOCK is set to 1, SM_OPEN is reset to 0 and further writes are prevented (read only). Additionally, further writes to SM_LOCK in this register, and HSM_RAM_EN, TSM_SIZE [1:0], and TSM_RAM_EN in the ESM_RAM_CR register are also prevented (read only). SM_LOCK can be set to 1 by a normal PCI configuration space write, but can only be cleared by a power-on reset. The BIOS can use SM_OPEN to initialize SMM space and then use SM_LOCK to lock down SMM space so that other software cannot violate the integrity of SMM space, even with knowledge of SM_OPEN.
3	GSM_RAM_EN	Global SMM RAM (SMRAM) Enable If this bit is set to 1, and the HSM_RAM_EN bit in the ESM_RAM_CR register is set to 0, then compatible SMRAM functions are enabled, providing 128 KB of DRAM accessible at the A0000h address while in SMM. To enable extended SMRAM function, this bit is set to 1. Refer to the SMM section for more details. When clear, all SMM memory is disabled. When set, SMM memory is enabled and the x86 address space location of SMM memory is determined by the HSM_RAM_EN and TSM_RAM_EN bits in the ESM_RAM_CR register. When SM_LOCK is set to 1, this bit becomes read-only.
2:0	CSM_BASE	Compatible SMM Space Base Segment This field indicates the location of SMM space. SMM DRAM is not remapped, it is simply made visible if the conditions are right to access SMM space. Otherwise, the access is forwarded to the PCI bus. This field is read-only, and is always set to the value '010' to indicate that the TM5700/TM5900 supports the compatible SMM space at A0000-BFFFFh.

73h Extended SMM RAM Control Register

Addr Offset	Register Name	Size	Access	Default Value
73h	ESM_RAM_CR	8 Bits	R/W	3Fh

The ESM_RAM_CR register controls the configuration of extended SMM RAM (SMRAM) space. The extended SMM RAM memory provides a write-back cacheable SMM RAM memory space that is above 1 MB.

ESM_RAM_CR Bit Definitions

Bit No.	Name	Function Description
7	HSM_RAM_EN	<p>High SMM RAM Enable The value of this bit controls the SMM memory space location (i.e. above 1 MByte or below 1 MByte).</p> <p>When GSM_RAM_EN and HSM_RAM_EN are both set to 1, the extended SMM memory space is enabled and the compatible SMM memory is disabled. Accesses in the 0A0000h to 0BFFFFh range are forwarded to the PCI bus, while SMRAM accesses from TOM + 1MB - 128K (TOM + 896K) to TOM + 1MB are enabled for SMM.</p> <p>When GSM_RAM_EN is set to 1 and HSM_RAM_EN is set to 0, the high SMM memory space is disabled and the compatible SMM memory from 0A0000h to 0BFFFFh is enabled.</p> <p>Once SM_LOCK is set, this bit becomes read only.</p>
6	--	Reserved.
5	SM_CACHE	<p>SMM RAM Cache Policy This bit is read only and always 1 to indicate a write-back cache policy for the TM5700/TM5900 SMM RAM.</p>
4	SM_L1_EN	<p>SMM RAM Cacheability This bit is read only and always 1 to indicate that both compatible and extended SMM RAM accesses are always cacheable by the TM5700/TM5900.</p>
3	SM_L2_EN	<p>SMM RAM Cacheability This bit is read only and always 1 to indicate that both compatible and extended SMM RAM accesses are always cacheable by the TM5700/TM5900.</p>
2:1	TSM_SIZE [1:0]	<p>T Segment SMM RAM Size Specifies the size of the SMM RAM T region, if enabled. This memory is taken from TM5700/TM5900 Code Morphing Software memory space (i.e. beyond TOM), which is no longer claimed by the TM5700/TM5900. The physical address for the extended SMM RAM memory appears in the range (TOM + 896K - TSM_SIZE [1:0]) to (TOM + 896K).</p> <p>These bits are read only and are set to '11' to indicate a T segment size of 896 KBytes (TOM to TOM + 896KB). All other bit combinations are reserved.</p> <p>The TM5700/TM5900 always allocates 896K for TSM_SIZE [1:0]. When additionally enabling high SMM (HSM_RAM_EN set to 1), this gives a full 1MB to the SMM handler for code and data.</p> <p>Once SM_LOCK is set, this bit becomes read only.</p>
0	TSM_RAM_EN	<p>T Segment SMM RAM Enable</p> <p>Enables SMRAM memory (T region, 896KB of additional SMRAM memory) for Extended SMRAM space only. When GSM_RAM_EN = 1 and TSM_RAM_EN = 1, the T region is enabled to appear in the appropriate physical address space.</p> <p>The TM5700/TM5900 always operates with TSM_RAM_EN = 1. This memory is always available for the SMM handler for code and data.</p>

78-79h ACPI Power Management Control Address Register

Addr Offset	Register Name	Size	Access	Default Value
78-79h	PM_CR2_ADDR	16 Bits	R/W	0022h

The address of the PM_CR2 register in I/O memory space.

PM_CR2_ADDR Bit Definitions

Bit No.	Name	Function Description
15:0	PMCR2ADDR	<p>ACPI Power Management Control Register (PM_CR2) Address This 16-bit value specifies the address of the PM_CR2 register in I/O space. The TM5700/TM5900 VNB allows relocating PM_CR2 to be compatible with other power management controller core logic components.</p> <p>For example, the ALI M1535 southbridge expects the PM_CR2 register at 30h in I/O space. This register defaults to 22h, which reflects the PM_CR2 register offset of the Intel 82443BX northbridge.</p>

7Ah Power Management Control Register

Addr Offset	Register Name	Size	Access	Default Value
7Ah	PM_CR	8 Bits	R/W and RO	18h

Power management control settings.

PM_CR Bit Definitions

Bit No.	Name	Function Description
7	--	Reserved.
6	PM_CR2_EN	<p>ACPI Control Register Enable Enables or disables access to the ACPI control register PM_CR2 (I/O address PM_CR2_ADDR, default is address 0022h).</p> <p>1: Enable. The ACPI control register is enabled, and all CPU cycles to the are handled by the TM5700/TM5900, i.e. not forwarded to PCI.</p> <p>0: Disable (default). All CPU cycles to PM_CR2 are forwarded to the PCI bus.</p>
5	--	Reserved.
4	NREF_EN	<p>Normal Refresh Enable This bit is read only and set to 1 to indicate that normal refresh operation is used following a Power On Suspend (POS) or Suspend to RAM (STR) state.</p>
3	QS_MODE	<p>Quick Start Mode This bit is read only and set to 1 to indicate that quick start mode is always enabled.</p>
2:0	--	Reserved.

7Bh ACPI Power Management Control Options Register

Addr Offset	Register Name	Size	Access	Default Value
7Bh	PM_CR2_OPT	8 Bits	R/W	00h

PM_CR2_OPT Bit Definitions

Bit No.	Name	Function Description
7	--	Reserved.
6	PM_CR2_FORWARD	Arbiter Control If set (1), I/O writes to the arbiter control port (as specified in PM_CR2_ADDR) are forwarded to the PCI bus (so that, e.g., the southbridge can snoop them). If clear (0), all I/O writes to the arbiter control port are handled internally by the TM5700/TM5900.
5:1	--	Reserved.
0	PSM_EN	Power Saving Mode Enable If set to (1), both the Code Morphing Software and the system SDRAM controllers (as present) enter a power saving mode during normal operation. In power saving mode, the clock enable lines to the Code Morphing Software and system memories are active only when they are being accessed or refreshed. This decreases power dissipation, but increases the latency for a memory cycle by one clock. Note that this power saving mode is orthogonal to the ACPI/APM CPU power saving (clock control) states. This option is disabled by default.

7Fh Power Management Control Register

Addr Offset	Register Name	Size	Access	Default Value
7Fh	PM_CONTROL	8 Bits	R/W	02h (see description)

Power management options register. For more details, see the entry for the pm_control field in the OEM Configuration Table chapter of the *Development and Manufacturing Guide*. Note that the default value is determined by the pm_control field.

PM_CONTROL Bit Definitions

Bit No.	Name	Function Description
7:2	--	Reserved.
1		Quick Start Mode If set to (1), put CD and SD memory into self-refresh in Quick Start. Higher latency, lower power. (default) If set to (0), do not put CD and SD memory into self-refresh in Quick Start. Lower latency, higher power.
0		DLL Reset Disable If set to (1), disable the DDR DDL reset when exiting self-refresh. If set to (0), do DDR DDL reset when exiting self-refresh which is normally performed. (default)

A0h Configuration Locks Register

Addr Offset	Register Name	Size	Access	Default Value
A0h	LOCK	8 Bits	R/WO	01h

The system BIOS will normally configure system memory. The LOCK register is used to allow the system manufacturer to prevent the BIOS from changing system memory timing parameters and sizing. LOCK is also used to prevent the BIOS from changing power management parameters. These bits become read only after they are set to 1.

LOCK Bit Definitions

Bit No.	Name	Function Description
7:3	--	Reserved.
2	LOCK_CD	Lock CD SDRAM Width Register Disables writes to the CD SDRAM WIDTH register (DRAM_WIDTH) when set. This bit is read-set, i.e. if 0 it can be turned to 1 but not vice versa.
1	LOCK_PM	Lock Power Management Control Registers When set, disables writes to the power management control registers (P_LVL2_*, P_LVL3_*, POS_*, and STR_*). Once set to 1, this bit becomes read-only.
0	--	Reserved.

A1h ROM Control Register

Addr Offset	Register Name	Size	Access	Default Value
A1h	ROMCTL	8 Bits	R/W*	06h (see description)

This VNB register is bi-modal. At any given time the register is in one of two states: direct (real), or virtual. The working state of the register is reflected in the read value of bit 7. The real state is signified by a setting of 0 in bit-7. The virtual state (VROMCTL) is signified by a setting of 1 in bit-7.

Note that the full writability of this register is gated by the value held by bit-0 of the 'upgrade_options' OEM config table field (the "secure" bit). This VNB register is fully writable if the secure-bit is 0; a value of 1 for the secure-bit implies that only bits [7:6] of the register can be written (placing the register in real or virtual access mode).

The bit definitions change depending on whether bit 7 is in direct ROMCTL mode (bit 7=0) or virtual mode (bit 7=1), as shown below.

ROMCTL Bit Definitions, Direct Mode (bit 7 = 0)

Bit No.	Function Description
7	Read as zero - this indicates that the register is in 'real' mode.
6	Read as zero; written as zero when trying to manipulate all other 7 bits of the register; written as one when the user is attempting to mask potential changes to bits [0:5]. That is, if you want to flip between virtual and real modes of access to this VNB register, you can do so in such a way that you preserve the state of the lower bits. To be clear, to switch between real and virtual access modes, you write 40h or C0h respectively to this register.
5	RNMI - this generally reads as one only if an 'upgrade & NMI trap' occurred. This is not the NMI of x86 space, but refers to a debugging mode of the Crusoe hardware. Writing this bit is harmless. One should expect this bit to read as zero, and always write it as zero.

ROMCTL Bit Definitions, Direct Mode (bit 7 = 0) (Continued)

Bit No.	Function Description
4	UPGRADE - this generally reads as one only if an 'upgrade & NMI trap' occurred as a direct result of a blocked attempt to write to the %romctl CSR. One should expect this bit to read as zero, and always write it as zero.
3	TOW - trap on write - this bit can be set to one and it has the effect that a subsequent write to this register will cause an 'upgrade & NMI trap' to be executed. This event causes the CPU to jump to its reset code again - executing out of the CMS ROM. This will likely cause the system to lock up. ALWAYS write this bit as zero, and expect it to be read as zero. Once the bit is read to be of value one, do not write to this register again, and do not try to access upgrade or persistent translation MSRs since any one of these events may cause the system to lock up.
2:1	The currently selected Flash ROM bank. CMS 4.5 supports four ROM banks, numbered 0 to 3.
0	The currently selected Flash ROM: 1=serial ROM; 0=parallel ROM.

ROMCTL Bit Definitions, Virtual Mode VROMCTL (bit 7 = 1)

Bit No.	Function Description
7	Read as one - this indicates that the register is in 'virtual' mode.
6	Read as zero; written as zero when trying to manipulate all other 7 bits of the register; written as one when the user is attempting to mask potential changes to bits [0:5]. That is, if you want to flip between virtual and real modes of access to this VNB register, you can do so in such a way that you preserve the state of the lower bits. To be clear, to switch between real and virtual access modes, you write 40h or C0h respectively to this register.
5	This bit indicates whether the GPIO pin is being driven high by the CPU or not. Its functionality is correct when bit-2 of the 'upgrade_options' OEM config table field is set to one.
4:3	Reserved as zero
2:1	The currently selected Flash ROM bank. CMS 4.5 supports four ROM banks, numbered 0 to 3.
0	The currently selected Flash ROM: 1=serial ROM; 0=parallel ROM.

A4-A7h OEM-Defined Meaning Register

Addr Offset	Register Name	Size	Access	Default Value
A4-A7h	OEMOPT	32 Bits	R/W	See description

OEMOPT Bit Definitions

Bit No.	Name	Function Description															
31:0	--	<p>OEM-Defined Field The properties of this register are defined by three fields in the OEM configuration table, as defined in the <i>OEM Configuration Application Note</i>: reset value, canset, and canclr.</p> <p>The OEMOPT default value is determined by the “reset value” field in the OEM configuration table.</p> <p>If a bit is set in the canset field in the OEM configuration table below, that particular bit can be set in the OEMOPT register by writing a 1 to the corresponding OEMOPT field. If a bit is set in the canclr field, that particular bit can be cleared in the OEMOPT register by writing a 0 to the corresponding OEMOPT field.</p> <p>OEM Configuration</p> <table border="1"> <thead> <tr> <th>canset</th> <th>canclr</th> <th>OEMOPT Access</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Read Only</td> </tr> <tr> <td>0</td> <td>1</td> <td>Read/Clear</td> </tr> <tr> <td>1</td> <td>0</td> <td>Read/Set</td> </tr> <tr> <td>1</td> <td>1</td> <td>Read/Write</td> </tr> </tbody> </table>	canset	canclr	OEMOPT Access	0	0	Read Only	0	1	Read/Clear	1	0	Read/Set	1	1	Read/Write
canset	canclr	OEMOPT Access															
0	0	Read Only															
0	1	Read/Clear															
1	0	Read/Set															
1	1	Read/Write															

A8h LongRun Advanced Thermal Management Definitions Register

Addr Offset	Register Name	Size	Access	Default Value
A8h	LR_ATM	8 Bits	R/W	01h

LR_ATM Bit Definitions

Bit No.	Name	Function Description
7:5	--	Reserved.
4	ATM_EN	<p>Advanced Thermal Management Enabled Settings are as follows:</p> <p>1: Enable LongRun advanced thermal management.</p> <p>0: Disable.</p>

LR_ATM Bit Definitions (Continued)

Bit No.	Name	Function Description																
3:1	ATM_LVL	<p>Advanced Thermal Management Level Selects the processor power reduction target. LongRun will attempt to meet or exceed the power reduction level specified in this register by lowering the processor frequency and voltage. This register has priority to reduce the processor frequency below that which is selected in the LongRun MSR.</p> <p>The setting in this register will never have the effect of increasing the processor frequency and voltage.</p> <p>Since the LongRun power reduction responds cubically to the corresponding performance reduction, it is significantly more efficient to implement thermal management with LongRun than with traditional clock throttling.</p> <p>The field is encoded as follows:</p> <table> <tr><td>000:</td><td>Reserved</td></tr> <tr><td>001:</td><td>Reserved</td></tr> <tr><td>010:</td><td>75.0% power reduction</td></tr> <tr><td>011:</td><td>62.5% power reduction</td></tr> <tr><td>100:</td><td>50.0% power reduction</td></tr> <tr><td>101:</td><td>37.5% power reduction</td></tr> <tr><td>110:</td><td>25.0% power reduction</td></tr> <tr><td>111:</td><td>12.5% power reduction</td></tr> </table> <p>Example: A value of 110 instructs LongRun to reduce the processor power by 25%, such that it will be limited to consume no more than 75% of its maximum power.</p>	000:	Reserved	001:	Reserved	010:	75.0% power reduction	011:	62.5% power reduction	100:	50.0% power reduction	101:	37.5% power reduction	110:	25.0% power reduction	111:	12.5% power reduction
000:	Reserved																	
001:	Reserved																	
010:	75.0% power reduction																	
011:	62.5% power reduction																	
100:	50.0% power reduction																	
101:	37.5% power reduction																	
110:	25.0% power reduction																	
111:	12.5% power reduction																	
0	LR_EN	<p>LongRun Enabled When this bit is set, the processor supports advanced thermal management via LongRun technology.</p>																

A9h**Processor Performance Control**

Addr Offset	Register Name	Size	Access	Default Value
A9h	PERF_CTRL	8 Bits	R/W	See description

Processor Performance Control register. Initial value is determined by the field “winxp_flags” in the OEM configuration table—see the *Development and Manufacturing Guide* for details.

PERF_CTRL Bit Definitions

Bit No.	Name	Function Description
7:1	--	Reserved.
0	PERF_CTRL	<p>Enable Windows XP Boot Performance Enhancements When set to 1, make performance trade-offs that favor the Windows XP boot workload. During normal operation (i.e. when not in the process of booting Windows XP), this bit should be 0 to make performance trade-offs that favor normal application workloads.</p>

AC-AFh PCI Arbiter Control

Addr Offset	Register Name	Size	Access	Default Value
AC-AFh	PCI_ARB_CTRL	32 bits	R/W	See description

This register defines priorities for the PCI arbiter as described below.

The default value is determined by determined by setting of OEM configuration table field `pci_arb_ctrl_initial_value` (see Chapter 3, *OEM Configuration Table*, in the *Development and Manufacturing Guide*). If `pci_arb_ctrl_initial_value` is 0, the initial value for `PCI_ARB_CTRL` is 002266A6h.

PCI_ARB_CTRL Bit Definitions

Bit No.	Name	Function Description																																				
31:24	--	Reserved																																				
23:21	SLOT7	An array of 8 3-bit priority descriptions. Each element (called a "slot") determines the relative priority of requests from the processor (internally generated requests), southbridge (PHOLD) and all other PCI devices (P_GNT#[5:0]) with respect to each other when determining which request will be honored if there are simultaneous requests pending. Each time the arbiter grants the bus, it advances to the next higher numbered slot wrapping around from SLOT7 to SLOT0.																																				
20:18	SLOT6																																					
17:15	SLOT5																																					
14:12	SLOT4																																					
11:9	SLOT3																																					
8:6	SLOT2		Priorities are defined as follows:																																			
5:3	SLOT1	<table border="1"> <thead> <tr> <th>Slot</th> <th>Highest</th> <th>Medium</th> <th>Lowest</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>Southbridge</td> <td>CPU</td> <td>PCI</td> </tr> <tr> <td>001</td> <td>PCI</td> <td>CPU</td> <td>Southbridge</td> </tr> <tr> <td>010</td> <td>Southbridge</td> <td>PCI</td> <td>CPU</td> </tr> <tr> <td>011</td> <td>PCI</td> <td>Southbridge</td> <td>CPU</td> </tr> <tr> <td>100</td> <td>CPU</td> <td>PCI</td> <td>Southbridge</td> </tr> <tr> <td>101*</td> <td>CPU</td> <td>Southbridge</td> <td>PCI</td> </tr> <tr> <td>110</td> <td>CPU</td> <td>Southbridge</td> <td>PCI</td> </tr> <tr> <td>111*</td> <td>CPU</td> <td>Southbridge</td> <td>PCI</td> </tr> </tbody> </table>	Slot	Highest	Medium	Lowest	000	Southbridge	CPU	PCI	001	PCI	CPU	Southbridge	010	Southbridge	PCI	CPU	011	PCI	Southbridge	CPU	100	CPU	PCI	Southbridge	101*	CPU	Southbridge	PCI	110	CPU	Southbridge	PCI	111*	CPU	Southbridge	PCI
Slot	Highest	Medium	Lowest																																			
000	Southbridge	CPU	PCI																																			
001	PCI	CPU	Southbridge																																			
010	Southbridge	PCI	CPU																																			
011	PCI	Southbridge	CPU																																			
100	CPU	PCI	Southbridge																																			
101*	CPU	Southbridge	PCI																																			
110	CPU	Southbridge	PCI																																			
111*	CPU	Southbridge	PCI																																			
2:0	SLOT0	* Value is reserved and should be avoided.																																				

D0-D7 BIOS Scratch Pad Register

Addr Offset	Register Name	Size	Access	Default Value
D0-D7	BSPAD	64 Bits	R/W	0000 0000 0000 0000h

This register provides 8 bytes of general purpose read/write registers that can be used by the BIOS as a workspace during configuration. The TM5700/TM5900 provides this 8 byte register in the PCI configuration space of the TM5700/TM5900 VNB device 0 on bus 0. The registers in this range will be defined as read/write and will be initialized to all 0's after PCIRST#. The BIOS can access these registers through the normal PCI configuration register mechanism, accessing 1, 2, or 4 bytes in each data access.

Several more BIOS scratch pad registers are also allocated. See *Extended BIOS Scratch Pad (Function 2)* on page 54.

BSPAD Bit Definitions

Bit No.	Name	Function Description
63:0	--	BIOS Work Space

D8-D9h Power Management Power-On Suspend Address Register

Addr Offset	Register Name	Size	Access	Default Value
D8-D9h	POS_ADDR	16 Bits	R/W	0000h

POS_ADDR Bit Definitions

Bit No.	Name	Function Description
15:0	POSADDR	Power Management Power On Suspend Monitor Address Specifies the I/O port address monitored to detect a transition to Power On Suspend mode.

DA-DBh Power Management Power-On Suspend Data Mask Register

Addr Offset	Register Name	Size	Access	Default Value
DA-DBh	POS_MASK	16 Bits	R/W	0000h

POS_MASK Bit Definitions

Bit No.	Name	Function Description
15:0	POSMASK	Power Management Power On Suspend Data Mask Specifies the mask value that is ANDed with the data read from or written to the monitor address.

DC-DDh Power Management Power-On Suspend Data Register

Addr Offset	Register Name	Size	Access	Default Value
DC-DDh	POS_DATA	16 Bits	R/W	0000h

POS_DATA Bit Definitions

Bit No.	Name	Function Description
15:0	POSDATA	Power Management Power On Suspend Monitor Data Specifies the value used when comparing the masked read or write data from the monitor address. If the two values are equal, the Power On Suspend transition is activated with the next STPCLK# assertion.

DEh Power Management Power-On Suspend Control Register

Addr Offset	Register Name	Size	Access	Default Value
DEh	POS_CTL	8 Bits	R/W	00h

POS_CTL Bit Definitions

Bit No.	Name	Function Description
7:2	--	Reserved.
1	POSWR	Power Management Power On Suspend Monitor on Write If set, the POS_ADDR port is monitored on write operations.
0	POSRD	Power Management Power On Suspend Monitor on Read If set, the POS_ADDR port is monitored on read operations.

E0-E1h Power Management Level 2 Address Register

Addr Offset	Register Name	Size	Access	Default Value
E0-E1h	P_LVL2_ADDR	16 Bits	R/W	0000h

P_LVL2_ADDR Bit Definitions

Bit No.	Name	Function Description
15:0	PL2ADDR	Power Management Level 2 Monitor Address Specifies I/O port address monitored to detect a transition to Quick Start mode.

E2-E3h Power Management Level 2 Data Mask Register

Addr Offset	Register Name	Size	Access	Default Value
E2-E3h	P_LVL2_MASK	16 Bits	R/W	0000h

P_LVL2_MASK Bit Definitions

Bit No.	Name	Function Description
15:0	PL2MASK	Power Management Level 2 Data Mask Specifies the mask value that is ANDed with the data read from or written to the monitor address.

E4-E5h Power Management Level 2 Compare Data Register

Addr Offset	Register Name	Size	Access	Default Value
E4-E5h	P_LVL2_DATA	16 Bits	R/W	0000h

P_LVL2_DATA Bit Definitions

Bit No.	Name	Function Description
15:0	PL2DATA	Power Management Level 2 Monitor Data Specifies the value used when comparing the masked read or write data from the monitor address. If the two values are equal, the power management level 2 transition is activated with the next STPCLK# assertion.

E6h Power Management Level 2 Control Register

Addr Offset	Register Name	Size	Access	Default Value
E6h	P_LVL2_CTL	8 Bits	R/W	00h

P_LVL2_CTL Bit Definitions

Bit No.	Name	Function Description
7:2	--	Reserved.
1	PL2WR	Power Management Level 2 Monitor on Write If set, the P_LVL2_ADDR port is monitored on write operations.
0	PL2RD	Power Management Level 2 Monitor on Read If set, the P_LVL2_ADDR port is monitored on read operations.

E8-E9h Power Management Level 3 Address Register

Addr Offset	Register Name	Size	Access	Default Value
E8-E9h	P_LVL3_ADDR	16 Bits	R/W	0000h

P_LVL3_ADDR Bit Definitions

Bit No.	Name	Function Description
15:0	PL3ADDR	Power Management Level 3 Monitor Address Specifies I/O port address monitored to detect a transition to Deep Sleep mode.

EA-EBh Power Management Level 3 Data Mask Register

Addr Offset	Register Name	Size	Access	Default Value
EA-EBh	P_LVL3_MASK	16 Bits	R/W	0000h

P_LVL3_MASK Bit Definitions

Bit No.	Name	Function Description
15:0	PL3MASK	Power Management Level 3 Data Mask Specifies the mask value that is ANDed with the data read from or written to the monitor address.

EC-EDh Power Management Level 3 Compare Data Register

Addr Offset	Register Name	Size	Access	Default Value
EC-EDh	P_LVL3_DATA	16 Bits	R/W	0000h

P_LVL3_DATA Bit Definitions

Bit No.	Name	Function Description
15:0	PL3DATA	Power Management Level 3 Monitor Data Specifies the value used when comparing the masked read or write data from the monitor address. If the two values are equal, the power management level 3 transition is activated with the next assertion.

EEh Power Management Level 3 Control Register

Addr Offset	Register Name	Size	Access	Default Value
EEh	P_LVL3_CTL	8 Bits	R/W	00h

P_LVL3_CTL Bit Definitions

Bit No.	Name	Function Description
7:2	--	Reserved.
1	PL3WR	Power Management Level 3 Monitor on Write If set, the P_LVL3_ADDR port is monitored on write operations.
0	PL3RD	Power Management Level 3 Monitor on Read If set, the P_LVL3_ADDR port is monitored on read operations.

F0-F1h Power Management Suspend to RAM Address Register

Addr Offset	Register Name	Size	Access	Default Value
F0-F1h	STR_ADDR	16 Bits	R/W	0000h

STR_ADDR Bit Definitions

Bit No.	Name	Function Description
15:0	STRADDR	Power Management Suspend to RAM Monitor Address Specifies the I/O port address monitored to detect a transition to the Suspend to RAM mode.

F2-F3h Power Management Suspend to RAM Data Mask Register

Addr Offset	Register Name	Size	Access	Default Value
F2-F3h	STR_MASK	16 Bits	R/W	0000h

STR_MASK Bit Definitions

Bit No.	Name	Function Description
15:0	STRMASK	Power Management Suspend to RAM Data Mask Specifies the mask value that is ANDed with the data read from or written to the monitor address.

F4-F5h Power Management Suspend to RAM Compare Data Register

Addr Offset	Register Name	Size	Access	Default Value
F4-F5h	STR_DATA	16 Bits	R/W	0000h

STR_DATA Bit Definitions

Bit No.	Name	Function Description
15:0	STRDATA	Power Management Suspend to RAM Monitor Data Specifies the value used when comparing the masked read or write data from the monitor address. If the two values are equal, the suspend to RAM transition is activated with the next STPCLK# assertion.

F6h Power Management Suspend to RAM Control Register

Addr Offset	Register Name	Size	Access	Default Value
F6h	STR_CTL	8 Bits	R/W	00h

STR_CTL Bit Definitions

Bit No.	Name	Function Description
7:2	--	Reserved.
1	STRWR	Power Management Suspend to RAM Monitor on Write If set, the STR_ADDR port is monitored on write operations.
0	STRRD	Power Management Suspend to RAM Monitor on Read If set, the STR_ADDR port is monitored on read operations.

FC-FDh Master I/O Clock Base Frequency Register

Addr Offset	Register Name	Size	Access	Default Value
FC-FDh	MASTER_CLK	16 Bits	RO	0000h

7

MASTER_CLK Bit Definitions

Bit No.	Name	Function Description
15:0	MASTER_CLOCK	Master Clock Base (CPU Core) Frequency The value of this register is the frequency in MHz of the TM5700/TM5900 processor core, as determined by the TM5700/TM5900 during initialization. This master clock frequency serves as the source for the SDRAM interface clocks and the PCI bus clock.

FEh Memory Clock Divisors Register

Addr Offset	Register Name	Size	Access	Default Value
FEh	MEMDIV	8 Bits	RO	00h

MEMDIV Bit Definitions

Bit No.	Name	Function Description
7:4	CD_DIV	Clock Divisor for Code-Morphing Software Memory These bits hold the value of the divisor used to divide the master clock (MASTER_CLK) base frequency down to the DDR SDRAM clock frequency, as determined by the TM5700/TM5900 during initialization. This register is read-only.
3:0	SD_DIV	SDR SDRAM Clock Divisor These bits hold the value of the divisor used to divide the master clock (MASTER_CLK) base frequency down to the SDR SDRAM clock frequency, as determined by the TM5700/TM5900 during initialization. This register is read-only.

FFh PCI Clock Divisor Register

Addr Offset	Register Name	Size	Access	Default Value
FFh	PCIDIV	8 Bits	RO	00h

PCIDIV Bit Definitions

Bit No.	Name	Function Description
7:0	PCI_DIV	PCI Clock Divisor This register holds the value of the divisor used to divide the master clock (MASTER_CLK) base frequency down to the PCI bus clock frequency, as determined by the TM5700/TM5900 during initialization. This register is read-only.

2.3.2 SDRAM Memory Controller Registers (Function 1)

This function of the VNB concerns memory control and initialization.

The SDRAM memory controller layout described in this section matches the SPD register layout as defined in the JEDEC Standard No. 21-C. This document reflects the Synchronous DRAM Modules with SPD revision level 2 (02h). Those PD's are those referenced in the SPD standard document for specific features. This standard can be found online at <http://www.jedec.org/download/pub21/>.

The most convenient way to program the SDRAM memory controller is by performing a block copy from the SPD ROM into the corresponding register bank in the SDRAM controller. Specifically, this means copying the first 64 bytes of the SPD ROM into function 1, starting at offset 60h for Bank 0, and at offset B0h for Bank 1.

Note

The Transmeta VNB only supports configuring both SDRAM bank 0 and bank 1 together, and they need to be setup in that exact order, first bank 0 and then bank 1.

Table 9: SDRAM Memory Controller Registers (PCI Function 1) Summary

Address Offset	Register Description	Register Name	Size	Access	Default Value	Page
00-01h	Vendor Identification Register	VID1	16 Bits	RO	1279h	51
02-03h	Device Identification Register	DID1	16 Bits	RO	0396h	51
08h	Revision Identification Register	RID1	8 Bits	RO	00h	51
09h	Programming Interface Register	PI1	8 Bits	RO	00h	52
0Ah	Sub-Class Code Register	SUBC1	8 Bits	RO	00h	52
0Bh	Base Class Code Register	BCC1	8 Bits	RO	05h	52
0Eh	Header Type Register	HEDT1	8 Bits	RO	80h	53
2C-2Dh	Subsystem Vendor Identification Register	SVID1	16 Bits	RO	See description	53
2E-2Fh	Subsystem Identification Register	SID1	16 Bits	RO	See description	53

00-01h Vendor Identification Register

Addr Offset	Register Name	Size	Access	Default Value
00-01h	VID1	16 Bits	RO	1279h

The VID1 register contains the vendor identification number. This register combined with the Device Identification Register (DID1) uniquely identify any PCI device. Writes to this register have no effect.

VID1 Bit Definitions

Bit No.	Name	Function Description
15:0	VID	Vendor Identification Number This 16-bit value is the PCI vendor identification number assigned to Transmeta. Transmeta VID = 1279h.

02-03h Device Identification Register

Addr Offset	Register Name	Size	Access	Default Value
02-03h	DID1	16 Bits	RO	0396h

This 16-bit register combined with the Vendor Identification Register (VID1) uniquely identifies any PCI device. Writes to this register have no effect.

DID1 Bit Definitions

Bit No.	Name	Function Description
15:0	DID	Device Identification Number This 16-bit value is the identification number assigned to the TM5700/TM5900 VNB function 1.

08h Revision Identification Register

Addr Offset	Register Name	Size	Access	Default Value
08h	RID1	8 Bits	RO	00h

The RID1 register contains the TM5700/TM5900 VNB function 1 revision identification number. Writes to this register have no effect.

RID1 Bit Definitions

Bit No.	Name	Function Description
7:0	RID	Revision Identification Number This 8-bit value is the revision identification number assigned to the TM5700/TM5900 power management controller function 1.

09h Programming Interface Register

Addr Offset	Register Name	Size	Access	Default Value
09h	PI1	8 Bits	RO	00h

Programming interface register, read-only.

PI1 Bit Definitions

Bit No.	Name	Function Description
7:0	PGIF	Programming Interface This read-only field always returns 0, since none of the Transmeta function 1 subclasses have further division into programming interfaces.

0Ah Sub-Class Code Register

Addr Offset	Register Name	Size	Access	Default Value
0Ah	SUBC1	8 Bits	RO	00h

The SUBC1 register contains the TM5700/TM5900 VNB category code. The default value is 00h, indicating a RAM device. For more information, see *PCI Hardware and Software—Architecture and Design (4th ed.)*, by Edward Solari and George Willse, Annabooks, San Diego, 1998, p. 562.

Writes to this register have no effect.

SUBC1 Bit Definitions

Bit No.	Name	Function Description
7:0	SUBC	Sub-Class Code This 8-bit value indicates the sub-class category type of memory controller that the TM5700/TM5900 PCI bridge function 1 provides. The default value 00h indicates a RAM device.

0Bh Base Class Code Register

Addr Offset	Register Name	Size	Access	Default Value
0Bh	BCC1	8 Bits	RO	05h

The BCC1 register contains the TM5700/TM5900 VNB function 1 base class code. The default value is 05h, indicating a memory controller. Writes to this register have no effect.

BCC1 Bit Definitions

Bit No.	Name	Function Description
7:0	BASEC	Base Class Code This 8-bit value indicates the base class code for the TM5700/TM5900 PCI bridge function 1. The default value is 05h, indicating a memory controller device.

0Eh Header Type Register

Addr Offset	Register Name	Size	Access	Default Value
0Eh	HEDT1	8 Bits	RO	80h

This register identifies the header layout of the configuration space. No physical register exists at this location.

HEDT1 Bit Definitions

Bit No.	Name	Function Description
7:0	HEADT	Header Type This 8-bit value indicates the header type for the TM5700/TM5900 PCI bridge function 1. The default value 80h is always returned by reads to this register. Bit 7 indicates that this device supports more than one function.

2C-2Dh Subsystem Vendor Identification Register

Addr Offset	Register Name	Size	Access	Default Value
2C-2Dh	SVID1	16 Bits	RO	See description

Subsystem vendor identification register.

SVID1 Bit Definitions

Bit No.	Name	Function Description
15:0	SVID	Subsystem Vendor ID This value is used to identify the vendor of the subsystem. It always reflects the value of the function 0 register SVID.

2E-2Fh Subsystem Identification Register

Addr Offset	Register Name	Size	Access	Default Value
2E-2Fh	SID1	16 Bits	RO	See description

Subsystem identification register.

SID1 Bit Definitions

Bit No.	Name	Function Description
15:0	SID	Subsystem ID This value is used to identify a particular subsystem. It always reflects the value of the function 0 register SID.

2.3.3 Extended BIOS Scratch Pad (Function 2)

This function of the VNB provides support for an extended BIOS scratch pad.

Table 10: Extended BIOS Scratch Pad Registers (PCI Function 2) Summary

Address Offset	Register	Register Name	Size	Access	Default Value	Page
00-01h	Vendor Identification Register	VID2	16 Bits	RO	1279h	54
02-03h	Device Identification Register	DID2	16 Bits	RO	0397h	54
08h	Revision Identification Register	RID2	8 Bits	RO	00h	55
09h	Programming Interface Register	PI2	8 Bits	RO	00h	55
0Ah	Sub Class Code Register	SUBC2	8 Bits	RO	00h	55
0Bh	Base Class Code Register	BCC2	8 Bits	RO	05h	56
0Eh	Header Type Register	HEDT2	8 Bits	RO	80h	56
2C-2Dh	Subsystem Vendor Identification Register	SVID2	16 Bits	RO	See description	56
2E-2Fh	Subsystem Identification Register	SID2	16 Bits	RO	See description	57
40-FFh	BIOS Scratch Pad Register Bank	BSPAD2[0:47]	32 Bits	R/W	0000 0000h	57

00-01h Vendor Identification Register

Addr Offset	Register Name	Size	Access	Default Value
00-01h	VID2	16 Bits	RO	1279h

The VID2 register contains the vendor identification number. This register combined with the Device Identification Register (DID2) uniquely identify any PCI device. Writes to this register have no effect.

VID2 Bit Definitions

Bit No.	Name	Function Description
15:0	VID	Vendor Identification Number This 16-bit value is the PCI vendor identification number assigned to Transmeta. Transmeta VID = 1279h.

02-03h Device Identification Register

Addr Offset	Register Name	Size	Access	Default Value
02-03h	DID2	16 Bits	RO	0397h

This 16-bit register combined with the Vendor Identification Register (VID2) uniquely identifies any PCI device. Writes to this register have no effect.

DID2 Bit Definitions

Bit No.	Name	Function Description
15:0	DID	Device Identification Number This 16-bit value is the identification number assigned to the TM5700/TM5900 VNB function 2.

08h Revision Identification Register

Addr Offset	Register Name	Size	Access	Default Value
08h	RID2	8 Bits	RO	00h

The RID2 register contains the TM5700/TM5900 VNB function 2 revision identification number. Writes to this register have no effect.

RID2 Bit Definitions

Bit No.	Name	Function Description
7:0	RID	Revision Identification Number This 8-bit value is the revision identification number assigned to the TM5700/TM5900 VNB function 2.

09h Programming Interface Register

Addr Offset	Register Name	Size	Access	Default Value
09h	PI2	8 Bits	RO	00h

Programming interface register, read-only.

PI2 Bit Definitions

Bit No.	Name	Function Description
7:0	PGIF	Programming Interface This read-only field always returns 0 when read, since none of the VNB function 2 subclasses have further division into programming interfaces.

0Ah Sub Class Code Register

Addr Offset	Register Name	Size	Access	Default Value
0Ah	SUBC2	8 Bits	RO	00h

The SUBC2 register contains the TM5700/TM5900 VNB category code. The default value is 00h, indicating a RAM device. For more information, see *PCI Hardware and Software - Architecture and Design (4th ed.)*, by Edward Solari and George Willse, Annabooks, San Diego, 1998, p. 562. Writes to this register have no effect.

SUBC2 Bit Definitions

Bit No.	Name	Function Description
7:0	SUBC	Sub-Class Code This 8-bit value indicates the sub-class category of memory controller that the TM5700/TM5900 VNB function 2 provides. The default value 00h indicates a RAM device.

0Bh Base Class Code Register

Addr Offset	Register Name	Size	Access	Default Value
0Bh	BCC2	8 Bits	RO	05h

The BCC2 register contains the TM5700/TM5900 VNB function 2 base class code. The default value is 05h, indicating a memory controller. For more information, see *PCI Hardware and Software—Architecture and Design (4th ed.)*, by Edward Solari and George Willse, Annabooks, San Diego, 1998, p. 562. Writes to this register have no effect.

BCC2 Bit Definitions

Bit No.	Name	Function Description
7:0	BASEC	Base Class Code This 8-bit value indicates the base class code for the TM5700/TM5900 PCI bridge function 2. The default value is 05h, indicating a memory controller device.

0Eh Header Type Register

Addr Offset	Register Name	Size	Access	Default Value
0Eh	HEDT2	8 Bits	RO	80h

This register identifies the header layout of the configuration space. No physical register exists at this location.

HEDT2 Bit Definitions

Bit No.	Name	Function Description
7:0	HEADT	Header Type This 8-bit value indicates the header type for the TM5700/TM5900 VNB function 2. The default value 80h is always returned by reads to this register. Bit 7 indicates that this device supports more than one function.

2C-2Dh Subsystem Vendor Identification Register

Addr Offset	Register Name	Size	Access	Default Value
2C-2Dh	SVID2	16 Bits	RO	See description

Subsystem vendor identification register.

SVID2 Bit Definitions

Bit No.	Name	Function Description
15:0	SVID	Subsystem Vendor ID This value is used to identify the vendor of the subsystem. It always reflects the value of the function 0 register SVID.

2E-2Fh Subsystem Identification Register

Addr Offset	Register Name	Size	Access	Default Value
2E-2Fh	SID2	16 Bits	RO	See description

Subsystem identification register.

SID2 Bit Definitions

Bit No.	Name	Function Description
15:0	SID	Subsystem ID This value is used to identify a particular subsystem. It always reflects the value of the function 0 register SID.

40-FFh BIOS Scratch Pad Register Bank

Addr Offset	Register Name	Size	Access	Default Value
40-FFh	BSPAD2[0:47]	32 Bits	R/W	0000 0000h

These 48 registers provide a 4-byte-sized extended general purpose read/write register bank for the BIOS to perform the configuration routine. The VNB provides these 4-byte registers in the PCI configuration space of the VNB Function 2 on bus 0. The registers in this range are read/write and are initialized to all 0's after PCIRST#. The BIOS can access these registers through the normal PCI configuration register mechanism, accessing 1, 2 or 4 bytes in every data access.

BSPAD2[0:47] Bit Definitions

Bit No.	Name	Function Description
15:0	--	BIOS Workspace

2.3.4 DDR SDRAM Memory Controller Registers (Func. 3)

The DDR SDRAM controller provides the interface to program the Primary SDRAM Width.

All registers not explicitly defined here are reserved and read-only. Registers at offset 50h-8Fh are consistent with the JEDEC SPD layout as defined in the JEDEC Standard No. 21-C, Annex D, DDR SDRAM, Rel. 11a.

Table 11: DDR SDRAM Memory Controller Registers (PCI Function 3) Summary

Address Offset	Register	Register Name	Size	Access	Default Value	Page
00-01h	Vendor Identification Register	VID3	16 Bits	RO	1279h	58
02-03h	Device Identification Register	DID3	16 Bits	RO	0399h	58
08h	Revision Identification Register	RID3	8 Bits	RO	00h	58
09h	Programming Interface Register	PI3	8 Bits	RO	00h	59
0Ah	Sub Class Code Register	SUBC3	8 Bits	RO	00h	59
0Bh	Base Class Code Register	BCC3	8 Bits	RO	05h	59
0Eh	Header Type Register	HEDT3	8 Bits	RO	80h	60
2C-2Dh	Subsystem Vendor Identification Register	SVID3	16 Bits	RO	See description	60

Table 11: DDR SDRAM Memory Controller Registers (PCI Function 3) Summary (Continued)

Address Offset	Register	Register Name	Size	Access	Default Value	Page
2E-2Fh	Subsystem Identification Register	SID3	16 Bits	RO	See description	60
5Dh	DRAM Width Register	DRAM_WIDTH	8 Bits	RW	See description	60
A8h-FFh	BIOS Scratch Pad Registers	BSPAD	32 Bits	RW	0000 0000h	61

00-01h Vendor Identification Register

Addr Offset	Register Name	Size	Access	Default Value
00-01h	VID3	16 Bits	RO	1279h

This register contains the vendor identification number. This register combined with the Device Identification Register (DID3) uniquely identify any PCI device. Writes to this register have no effect.

VID3 Bit Definitions

Bit No.	Name	Function Description
15:0	VID	Vendor Identification Number This 16-bit value is the PCI vendor identification number assigned to Transmeta. Transmeta VID = 1279h.

02-03h Device Identification Register

Addr Offset	Register Name	Size	Access	Default Value
02-03h	DID3	16 Bits	RO	0399h

This 16-bit register combined with the Vendor Identification Register (VID3) uniquely identifies any PCI device. Writes to this register have no effect.

DID3 Bit Definitions

Bit No.	Name	Function Description
15:0	DID	Device Identification Number This 16-bit value is the identification number assigned to the TM5700/TM5900 VNB function 3.

08h Revision Identification Register

Addr Offset	Register Name	Size	Access	Default Value
08h	RID3	8 Bits	RO	00h

The RID3 register contains the TM5700/TM5900 VNB function 3 revision identification number. Writes to this register have no effect.

RID3 Bit Definitions

Bit No.	Name	Function Description
7:0	RID	Revision Identification Number This 8-bit value is the revision identification number assigned to the TM5700/TM5900 VNB function 3.

09h Programming Interface Register

Addr Offset	Register Name	Size	Access	Default Value
09h	PI3	8 Bits	RO	00h

Programming interface register, read-only.

PI3 Bit Definitions

Bit No.	Name	Function Description
7:0	PGIF	Programming Interface This read-only field always returns 0 when read, since none of the VNB function 3 subclasses have further division into programming interfaces.

0Ah Sub Class Code Register

Addr Offset	Register Name	Size	Access	Default Value
0Ah	SUBC3	8 Bits	RO	00h

The SUBC3 register contains the TM5700/TM5900 VNB category code. The default value is 00h, indicating a RAM device. For more information, see *PCI Hardware and Software - Architecture and Design (4th ed.)*, by Edward Solari and George Willse, Annabooks, San Diego, 1998, p. 562. Writes to this register have no effect.

SUBC3 Bit Definitions

Bit No.	Name	Function Description
7:0	SUBC	Sub-Class Code This 8-bit value indicates the sub-class category of memory controller that the TM5700/TM5900 VNB function 3 provides. The default value 00h indicates a RAM device.

0Bh Base Class Code Register

Addr Offset	Register Name	Size	Access	Default Value
0Bh	BCC3	8 Bits	RO	05h

The BCC3 register contains the TM5700/TM5900 VNB function 3base class code. The default value is 05h, indicating a memory controller. For more information, see *PCI Hardware and Software—Architecture and Design (4th ed.)*, by Edward Solari and George Willse, Annabooks, San Diego, 1998, p. 562. Writes to this register have no effect.

BCC3 Bit Definitions

Bit No.	Name	Function Description
7:0	BASEC	Base Class Code This 8-bit value indicates the base class code for the TM5700/TM5900 PCI bridge function 3. The default value is 05h, indicating a memory controller device.

0Eh Header Type Register

Addr Offset	Register Name	Size	Access	Default Value
0Eh	HEDT3	8 Bits	RO	80h

This register identifies the header layout of the configuration space. No physical register exists at this location.

HEDT3 Bit Definitions

Bit No.	Name	Function Description
7:0	HEADT	Header Type This 8-bit value indicates the header type for the TM5700/TM5900 VNB function 3. The default value 80h is always returned by reads to this register. Bit 7 indicates that this device supports more than one function.

2C-2Dh Subsystem Vendor Identification Register

Addr Offset	Register Name	Size	Access	Default Value
2C-2Dh	SVID3	16 Bits	RO	See description

Subsystem vendor identification register.

SVID3 Bit Definitions

Bit No.	Name	Function Description
15:0	SVID	Subsystem Vendor ID This value is used to identify the vendor of the subsystem. It always reflects the value of the function 0 register SVID.

2E-2Fh Subsystem Identification Register

Addr Offset	Register Name	Size	Access	Default Value
2E-2Fh	SID3	16 Bits	RO	See description

Subsystem identification register.

SID3 Bit Definitions

Bit No.	Name	Function Description
15:0	SID	Subsystem ID This value is used to identify a particular subsystem. It always reflects the value of the function 0 register SID.

5Dh DRAM Width Register

Addr Offset	Register Name	Size	Access	Default Value
5Dh	DRAM_WIDTH	8 Bits	RW	See description

When this register is written, the VNB recalculates the LongRun memory speeds based on the number of loads computed from this register and other internal state. Writing a value of zero to this register causes the

VNB to use the SDRAM Width field from the OEM configuration table, in which case this register is be set to the OEM configuration table's value.

The default value for this register is the SPD ROM programmed in the OEM configuration table.

Note that writes to this register are disabled if the LOCK_CD_WIDTH bit has been set in the LOCK register (see page 39)..

DRAM_WIDTH Bit Definitions

Bit No.	Function Description
7:0	Primary SDRAM Width This register defines the primary data SDRAM width. Bits 6:0 define the primary data SDRAM width. The TM5900 Virtual Northbridge supports x4, x8, x16 and x32 primary data widths. Bit 7 is zero if only one physical bank is present or if two physical banks with the same primary data widths are present on the DIMM module. Bit 7 is a 1 if the second physical bank has a primary data width different from physical bank 1. The second physical bank width must be equal to physical bank 1 or 2X physical bank 1.

A8h-FFh BIOS Scratch Pad Registers

Addr Offset	Register Name	Size	Access	Default Value
A8h-FFh	BSPAD	32 Bits	RW	0000 0000h

This block of 22 DWORD registers provide an extended general-purpose read/write register bank for the BIOS to perform the configuration routine. The BIOS can access these registers through the normal PCI configuration register mechanism, accessing 1, 2 or 4 bytes in every data access. .

BSPAD Bit Definitions

Bit No.	Name	Function Description
15:0	SID	Extended BIOS Workspace

Model-Specific Registers

This chapter describes the Model-Specific Registers (MSRs) that are supported by Transmeta processors, as well as the instructions that can be used to access those MSRs.

3.1 Introduction

The Model-Specific Registers allow you to observe and control the behavior of processor-specific details. Three instructions can access the MSRs, and the CPUID instruction can be used to determine whether MSRs are supported or not. Crusoe processors do support MSRs, including the timestamp counter (TSC).

In general, MSRs are 64 bits wide, though not every MSR implements all 64 bits. Reserved bits read as 0, while writing a 1 into a reserved bit generates a GP(0) exception. The RDMSR and WRMSR instructions read and write MSR values, while the RDTSC instruction reads the TSC value. Trying to read from or write to a reserved MSR generates a GP(0) exception.

The RDMSR and WRMSR instructions are privileged, and WRMSR is serializing. Both of these instructions require the MSR index as an input value, in register ECX. The RDMSR instruction returns the current MSR value in EDX (high-order 32 bits) and EAX (low-order 32 bits). WRMSR requires the desired MSR value as an input, also in EDX and EAX as described.

Because the MSRs differ from processor to processor, your programs should use them with caution.

3.2 Intel-Compatible MSRs

The following Intel-compatible MSRs are provided. All are 64 bits wide.

Table 12: Intel-Compatible MSR Summary

Index	Register	Register Name	Access	Page
0000 0010h	Timestamp Counter Register	TSC	R/W	64
0000 0119h	Processor Serial Number Disable Register	PSN_DISABLE	R/WO	64

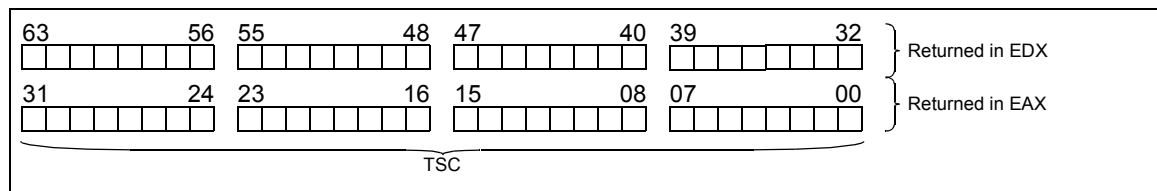
Timestamp Counter Register

Index	Register Name	Access
0000 0010h	TSC	R/W

This 64-bit register increments once for every processor core clock cycle. If the processor core clock frequency changes, the increment rate of the TSC reflects that change.

The RDTSC instruction is only privileged if CR4.TSD is set to 1. It expects no input values, and returns the current TSC value in EDX and EAX. Do not use RDMSR to read the TSC, as the actual MSR that holds the TSC might be implemented at a different index in the future. Moreover, if you use the TSC for instruction timing, you must keep the non-deterministic instruction timing of Crusoe processors in mind.

TSC Bit Definitions



Processor Serial Number Disable Register

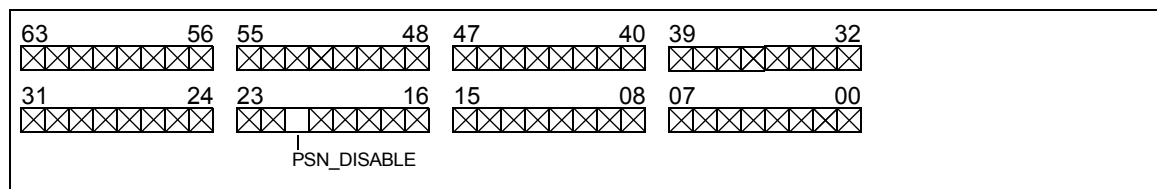
Index	Register Name	Access
0000 0119h	PSN_DISABLE	R/WO

Use this register to disable the processor serial number.

Set bit 21 to 1 to disable the PSN. The bit cannot be cleared once it is set to 1, except by processor RESET (see Appendix C, *Initial Processor State*).

The value of bit 21 after RESET is determined by the configuration variable `psn_disable`. For more information, see the *Development and Manufacturing Guide*.

PSN_DISABLE Bit Definitions



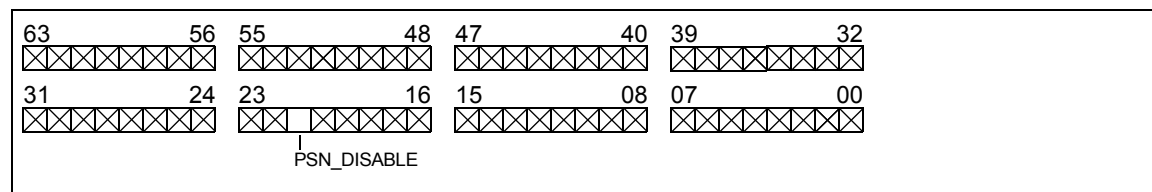
SYSENTER/SYSEXIT Base Selector

Index	Register Name	Access
0000 0174h	SEP_SEL	RO

Support for this MSR is indicated by a standard feature flag (SEP) returned by the CPUID instruction.

This is the 16-bit selector that is used by the SYSENTER and SYSEXIT instructions. The upper 32 bits are ignored during WRMSR, and always read as zero. Note that bits 31-16 are scratch bits—they hold their value but have no function.

SEP_SEL Bit Definitions



SYSENTER ESP

Index	Register Name	Access
0000 0175h	SEP_ESP	RO

Support for this MSR is indicated by a standard feature flag (SEP) returned by the CPUID instruction.

This is the 32-bit ESP value that is used by the SYSENTER instruction. The upper 32 bits are ignored during WRMSR, and always read as zero.

SYSENTER EIP

Index	Register Name	Access
0000 0176h	SEP_EIP	RO

Support for this MSR is indicated by a standard feature flag (SEP) returned by the CPUID instruction.

This is the 32-bit EIP value that is used by the SYSENTER instruction. The upper 32 bits are ignored during WRMSR, and always read as zero.

3.3 Transmeta-Specific MSRs

The following Transmeta-specific MSRs are provided:

Table 13: Transmeta-Specific MSR Summary

Index	Register	Register Name	Access	Pg
8086 0000h	CPUID Type, Family, Model, and Stepping Register	CPUID_TFMS	R/W	66
8086 0001h	CPUID Vendor ID String Registers	CPUID_VND1	R/W	67
8086 0002h		CPUID_VND2		
8086 0003h		CPUID_VND3		
8086 0004h	CPUID Feature Flag Mask Registers	CPUID_MASK	R/W	67

Table 13: Transmeta-Specific MSR Summary (Continued)

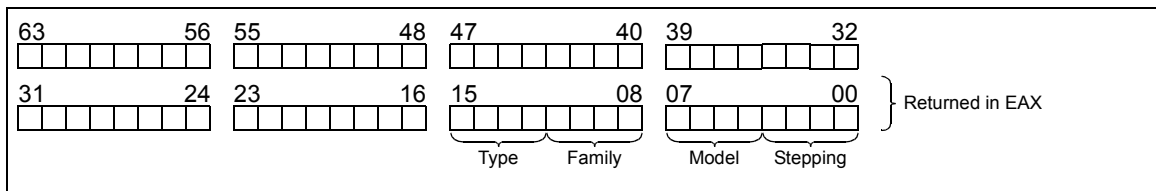
Index	Register	Register Name	Access	Pg
8086 8010h	LongRun™ Control and Status Register	LONGRUN	R/W	68
8086 0074h	CMS Boot Revision and Build Number Register	BOOT_REVISION_ID	R/W	69

CPUID Type, Family, Model, and Stepping Register

Index	Register Name	Access
8086 0000h	CPUID_TFMS	R/W

This MSR allows you to specify the type, family, model, and stepping values that are reported by the CPUID instruction for function 0000 0001h in register EAX (see *CPUID Standard Functions* on page 11 in Chapter 1, *Processor and Frequency Detection*). Each of these values must be between 0h and Fh.

CPUID_TFMS Bit Definitions



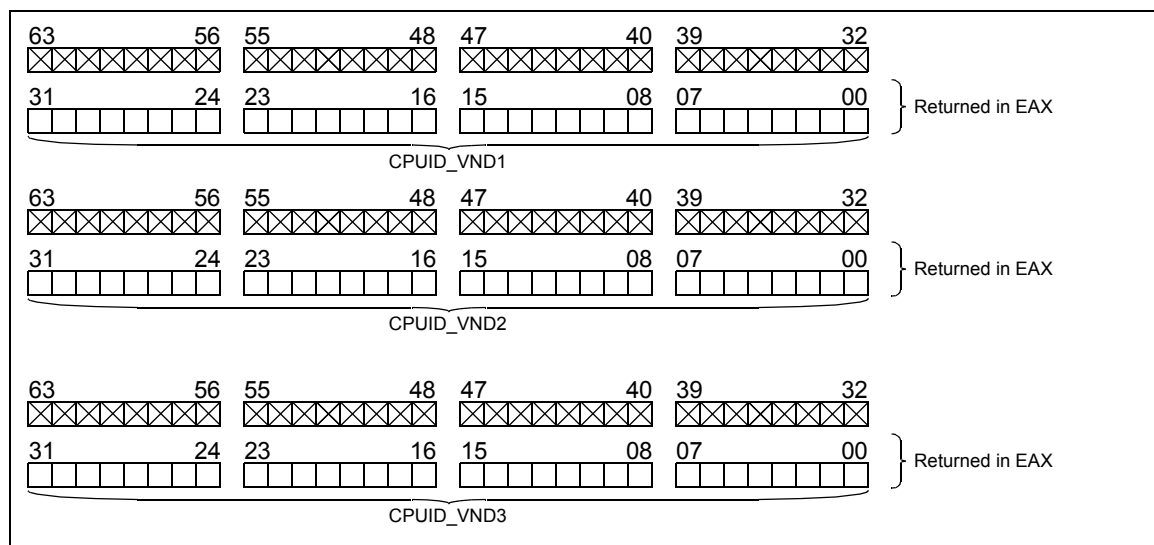
CPUID Vendor ID String Registers

Index	Register Name	Access	Report to Register... ¹
8086 0001h	CPUID_VND1	R/W	EAX
8086 0002h	CPUID_VND2		EAX
8086 0003h	CPUID_VND3		EAX

1. Lower 32 bits only are reported in the corresponding register.

These three MSR's allow you to specify the vendor ID string reported by the CPUID instruction for function 0000 0000h in register EAX. The MSR's form a contiguous string and map to the CPUID output registers in order, as shown in the diagram below. Note that this is different from the similar CPUID function.

CPUID_VNDn Bit Definitions



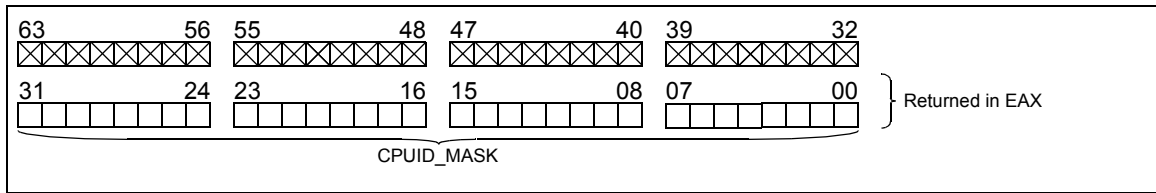
CPUID Feature Flag Mask Registers

Index	Register Name	Access
8086 0004h	CPUID_MASK	R/W

This MSR allows you to mask out the feature flags reported by the CPUID instruction for function 0000 0001h in register EDX (see *CPUID Standard Functions* on page 11 in Chapter 1, *Processor and Frequency Detection*). If a bit is set to 0 in this mask, then the corresponding feature flag is forced to 0; otherwise the corresponding feature flag is reported without changes.

Bit 8, indicating CMPXCHG support, is masked by default in older versions of CMS (before 4.2.x). CMPXCHG support is enabled in current versions of CMS to support Windows XP.

CPUID_MASK Bit Definitions



LongRun™ Control and Status Register

Index	Register Name	Access
8086 8010h	LONGRUN	R/W

When LongRun is available, this MSR allows you to select the processor's performance window, and query the current performance window. Use CPUID function 8086 0001h to test whether LongRun support is available. Additional LongRun status information can be queried via CPUID function 8086 0007h. Both of these functions are described in Chapter 1, *Processor and Frequency Detection*, in the section *CPUID Transmeta-Specific Functions* on page 15.

Bits 63..32 hold a number which must be no larger than 0000 0064h (100 decimal, or 100%); this number is the upper boundary of the performance window, returned in register EDX on RDMSR.

Bits 31..0 hold a number which must be no larger than 0000 0064h (100 decimal, or 100%); this number is the lower boundary of the performance window, returned in register EAX on RDMSR.

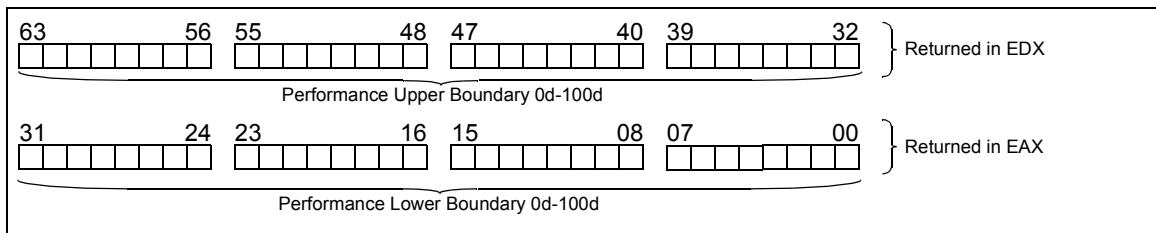
Use WRMSR to specify the desired performance window. Specifying boundary values outside the range 0000 0000h–0000 0064h causes a GP# fault, as does specifying a lower boundary that is greater than the upper boundary.

A performance window can be achieved by specifying an upper and lower boundary. A fixed performance level can be achieved by specifying the same value for the upper and lower boundary. Setting both boundaries to 0000 0064h disables LongRun and results in maximum performance, but short battery lifetime. By contrast, setting both boundaries to 0 results in minimum performance, but best battery lifetime.

The processor will round the specified values up or down to the nearest value, depending on how many performance levels are supported. If both directions of rounding are possible, then the processor rounds up.

Use RDMSR with this MSR to query the current performance window. Notice that the processor returns rounded values; thus they may or may not match the most recently written values. The upper boundary is returned in EDX, the lower boundary is returned in EAX.

CPUID_MASK Bit Definitions

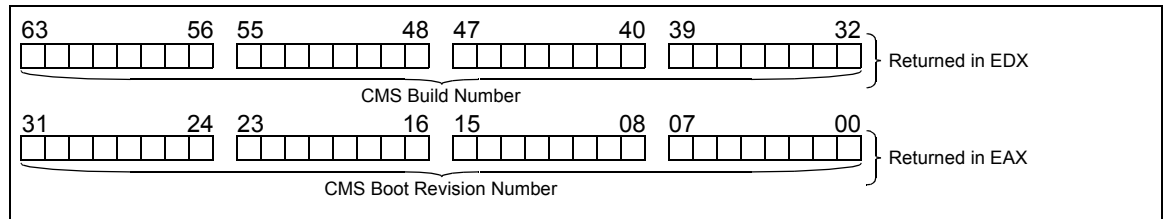


CMS Boot Revision and Build Number Register

Index	Register Name	Access
8086 0074h	BOOT_REVISION_ID	R/W

This MSR returns the Code Morphing Software boot revision and build number. On read, EAX shows the revision number, and EDX shows the build number.

xxxx Bit Definitions



Power Management

4.1 Power Management Control

The TM5700/TM5900 supports ACPI-compliant power management modes. The following table lists the recommended state of the processor for each of the ACPI global system states. The states are described in *Processor Power States* on page 72.

Table 14: Power Management System States

ACPI System State	TM5700/TM5900 State	DDR, SDR SDRAM	Clock Generator
G0/C0: Working	Normal	Normal	Running
G0/C1: Working/Auto Halt	Normal/ AutoHalt	Normal	Running
G0/C2: Working/Quick Start	Normal/ Quick Start	Normal/ Self refresh	Running
G1/S1/C1: Sleeping/Auto Halt	Auto Halt	Normal	Running
G1/S1/C2: Sleeping/Quick Start	Quick Start	Self refresh	Running
G1/S1/C3: Sleeping/Deep Sleep	Deep Sleep	Self refresh	CLKIN stopped
G1/S3: Sleeping/Suspend to RAM	Off	Self refresh	All clocks stopped
G1/S4: Sleeping/Suspend to disk	Off	Off	Off
G2/S5: Soft off	Off	Off	Off
G3: Mechanical off	Off	Off	Off

4.1.1 Processor Power States

The TM5700/TM5900 supports the following processor power-saving states:

Table 15: Processor Power States

State	ACPI Power State	Description
Auto Halt	C1	Auto Halt is normally initiated via execution of the HLT instruction.
Quick Start Throttling	C2	Quick Start Throttling is achieved by periodic assertion and deassertion of the STPCLK# signal. Throttling is normally configured through the southbridge. The TM5700/TM5900 supports asynchronous throttling events, such as throttling in response to a thermal event. There are no northbridge registers immediately affected by this power state.
Quick Start	C2	Quick Start is initiated upon assertion of the STPCLK# signal. This is normally accomplished by issuing a command to the southbridge. The Transmeta VNB LVL2 snooper must be configured to recognize the command that initiates Quick Start. See <i>State Transitions</i> below. Registers affected: P_LVL2_ADDR, P_LVL2_MASK, P_LVL2_DATA, P_LVL2_CTL, PM_CR2
Deep Sleep	C3	Deep Sleep is initiated upon assertion of the STPCLK# signal, followed by assertion of the SLP# signal. Host clocks can be stopped in the Deep Sleep state. It is recommended that you disable the PCI arbiter before entering Deep Sleep. The final power-state transition is normally triggered by a software command to the southbridge. The Transmeta VNB LVL3 snooper must be configured to recognize the command that initiates Deep Sleep. See <i>State Transitions</i> below. Registers affected: P_LVL3_ADDR, P_LVL3_MASK, P_LVL3_DATA, P_LVL3_CTL, PM_CR2, PM_CR

Table 15: Processor Power States (Continued)

State	ACPI Power State	Description
Suspend to RAM	S3	<p>Suspend to RAM (STR) cuts off power from most of the system components except DRAM. We recommend that you disable the PCI arbiter before entering STR. The sequence for initiating STR is complex and can vary from system to system. The final power state transition is normally triggered by a software command to the southbridge.</p> <p>The Transmeta VNB STR snooper must be configured to recognize the command that initiates Suspend to RAM. See <i>State Transitions</i> below.</p> <p>Registers affected: STR_ADDR, STR_MASK, STR_DATA, STR_CTL, PM_CR2, PM_CR</p>
Power On Suspend	S1	<p>Power On Suspend is initiated upon assertion of the STPCLK# signal, followed by assertion of the SLP# signal. Host clocks and the PCI clock are usually stopped in the Power On Suspend state. It is recommended that you disable the PCI arbiter before entering Power On Suspend. The final power-state transition is normally triggered by a software command to the southbridge.</p> <p>The Transmeta VNB POS snooper must be configured to recognize the command that initiates Power On Suspend. See <i>State Transitions</i> below.</p> <p>Registers affected: POS_ADDR, POS_MASK, POS_DATA, POS_CTL, PM_CR2, PM_CR</p>

4.1.2 State Transitions

Power management for the Transmeta processor and virtual northbridge is unique with respect to power-state transitions. Most chipsets initiate power state transitions using only the southbridge. The northbridge and processor are usually ignorant of the finer details of power management. The TM5700/TM5900 provides superior power management by increasing processor and northbridge awareness of power state transitions.

The northbridge snoopers can be programmed to recognize the commands that initiate the LVL2 (Quick Start), LVL3 (Deep Sleep), and STR (Suspend to RAM) power states. This method enables the processor and northbridge to be fully aware of power state transitions, while allowing system designers to fully leverage existing southbridge-centric power management code.

4.1.3 State Preservation

The Transmeta VNB preserves its state during the ACPI S1-S2 system sleep modes, and specifically during Suspend to RAM (STR). That is, upon resume from STR, the BIOS (or OS) isn't required to restore the registers (or state) of the Transmeta VNB, because they automatically reflect the state before the STR.

4.2 Power Management Registers

BIOS typically controls entry/exit to and from these power management states via registers in the southbridge device on the base platform. In addition to controlling the southbridge registers, BIOS must initialize power management registers in the TM5700/TM5900's PCI configuration space (addresses D8-F7h). These registers enable the TM5700/TM5900 to properly prepare for upcoming power management state transitions.

The power management registers are divided into four sets:

Registers	Address Offset	Description
P_LVL2_*	E0-E6h	Specifies the trigger event for a transition to the Quick Start state.
P_LVL3_*	E8-EEh	Specifies the trigger event for a transition to the Deep Sleep state.
STR_*	F0-F6h	Specifies the trigger event for a transition to the Suspend to RAM state.
POS_*	D8-DEh	Specifies the trigger event for a transition to the Power On Suspend state.

Typically, the trigger events for these power level transitions are reads or writes to I/O registers in the southbridge.

Each set of registers contains four components, as described below:

Register	Description
*_ADDR	Address The P_LVL2_ADDR, P_LVL3_ADDR, and STR_ADDR registers contain the I/O address of the state transition command. NOTE: The address must be configured to the exact address that the state transition software accesses to trigger the state transition.
*_MASK	Data Mask The P_LVL2_MASK, P_LVL3_MASK, and STR_MASK registers contain the data mask for the state transition command. Only bit positions which are set to 1 in the mask are evaluated against the data value to determine whether the I/O cycle triggers the state transition.
*_DATA	Data The P_LVL2_DATA, P_LVL3_DATA, and STR_DATA registers contain the data value of the state transition command. Only bit positions which are set to 1 in the corresponding mask are evaluated against the data value to recognize the intended command.
*_CTL	I/O Access Control. The P_LVL2_CTL, P_LVL3_CTL, and STR_CTL registers control which type of I/O access (read, write, or both) can trigger the power state transition. Setting this register to 0 disables the snooper.

4.3 LongRun™ Power Management

The TM5700/TM5900 incorporates LongRun power management, an advanced form of processor power management. LongRun utilizes adaptive voltage and clock scaling to match the processor performance and energy consumption levels to the required workload.

LongRun is enabled using the CPUID instruction. For more information, see *CPUID Transmeta-Specific Functions* on page 15. LongRun is configured using the LONGRUN model-specific register. For more information, see *LongRun™ Control and Status Register* on page 68.

4.4 Example Configuration for PIIX4

The following lists example settings for the power management registers assuming a PIIX4 southbridge is used to trigger the power level transitions. The PIIX4 power management registers are located in the PCI Configuration Register space for Function 3.

Table 16: Example Settings for Power Management Registers (PIIX4 Southbridge)

Register	Value	Description
P_LVL2_ADDR	PMBA + 14h	I/O address of the PLVL2 register in the PIIX4. NOTE: Assumes Stop Clock Enable bit is set in the PCNTRL register in the PIIX4. This bit enables the hardware signal that is used to transition the processor into Quick Start.
P_LVL2_MASK	0000h	Default value. No data check required.
P_LVL2_DATA	0000h	Default value. No data check required.
P_LVL2_CTL	01h	Monitor the PLVL2 register only on read operations.
P_LVL3_ADDR	PMBA + 15h	I/O address of the PLVL3 register in the PIIX4. NOTE: Assumes Stop Clock Enable, Sleep Enable and Clock Control Enable bits are set in the PCNTRL register in the PIIX4. These bits enable the hardware signals that are used to transition the processor into the Deep Sleep state.
P_LVL3_MASK	0000h	Default value. No data check required.
P_LVL3_DATA	0000h	Default value. No data check required.
P_LVL3_CTL	01h	Monitor the PLVL3 register only on read operations.
STR_ADDR	PMBA + 04h	I/O address of the PMCNTRL register in the PIIX4.
STR_MASK	3C00h	Unmask bits 13:10 of the PMCNTRL register.
STR_DATA	2400h	Check for SUS_EN enabled and SUS_TYP = suspend to RAM.
STR_CTL	02h	Monitor the PMCNTRL register only on write operations.
POS_ADDR	PMBA + 04h	I/O address of the PMCNTRL register in the PIIX4.
POS_MASK	3C00h	Unmask bits 13:10 of the (PMCNTRL) register.
POS_DATA	3000h	Check for SUS_EN enabled and SUS_TYP = power on suspend, context maintained.
POS_CTL	02h	Monitor the (PMCNTRL) register only on write operations.

4.5 Special Considerations

4.5.1 PCI Arbiter

The PCI arbiter should be disabled for all sleep states deeper than Quick Start. This can be accomplished by setting the ARB_DIS bit in the PM_CR2 I/O register (port 22h, bit 0). The PM_CR2_EN bit in the PM_CR register (PCI register 7Ah, bit 6) must be set prior to accessing the PM_CR2 register.

4.5.2 Configuration Register Context during STR

The memory attribute registers (PAB0-PAB6) are reset during the Suspend to RAM (STR) power management state. The value of all other configuration registers is automatically preserved.

4.5.3 DRAM Refresh

The self refresh method of DRAM refresh is only supported when clocks are removed from the SDRAM. All actions necessary to initiate self refresh are handled internally by the processor. No additional software steps are required.

4.5.4 STPCLK Behavior of ALI 1535

Set bit 3 (enable clock control don't check break event in advance) in the MISC_CNTL2 (0xa7) register of the PMU device of the ALI 1535 southbridge. This allows the southbridge to assert—and then immediately deassert—STPCLK in the case that there is already a pending break event when the STPCLK is requested.

If this bit is not set, the STPCLK is postponed until the break event has been cleared. This can lead to a situation where the STPCLK is delivered at a time when no possible break events can occur and so the system never wakes up.

This issue is specific to the ALI 1535. Note that the issue applies to any ACPI-aware operating system.

System Management Mode

The TM5700/TM5900 supports System Management Mode (SMM). SMM is intended for use only by system firmware and is typically used for system power management functions. SMM offers a distinct processor operating environment that is transparent to the operating system and software applications.

5.1 SMM Address Space

SMM address space should be initialized prior to executing the system management interrupt (SMI) handler code. The TM5700/TM5900 provides two different SMM memory configurations: standard SMM memory configuration, and high SMM memory configuration.

5.1.1 Standard and Extended SMM Memory Configuration

The standard SMM memory configuration consists of two blocks of SMM memory that may be used while in SMM:

- standard SMM address space
- extended SMM address space

Standard SMM address space is 128 KBytes of SMM RAM that physically resides in SDRAM from (TOM + 896 KBytes) to (TOM + 1 MByte). The standard SMM memory is accessed using addresses in the A0000h-BFFFFh range which are then redirected to the physical memory above TOM. The System Management RAM Control Register (SM_RAM_CR), and the Extended SMM RAM Control Register (ESM_RAM_CR) provide control bits for enabling/initializing standard SMM memory, and redirecting SMM accesses.

Extended SMM address space is 896 KBytes of SMM RAM that physically resides in SDRAM from (TOM) to (TOM + 896 KBytes). This memory is accessed using the actual physical addresses. The System Management RAM Control Register (SM_RAM_CR), and the Extended SMM RAM Control Register (ESM_RAM_CR) provide control bits for enabling and initializing extended SMM memory.

When using the standard SMM memory configuration, the video frame buffer (PCI memory A0000h to BFFFFh) is displaced by standard SMM memory. Therefore, the video frame buffer must be accessed

through an alternate address space as defined by the Virtual Video Window Base Address Register (VWBASE). The alternate address space extends from (VWBASE + A0000h) to (VWBASE+ BFFFFh).

5.1.2 High SMM Memory Configuration

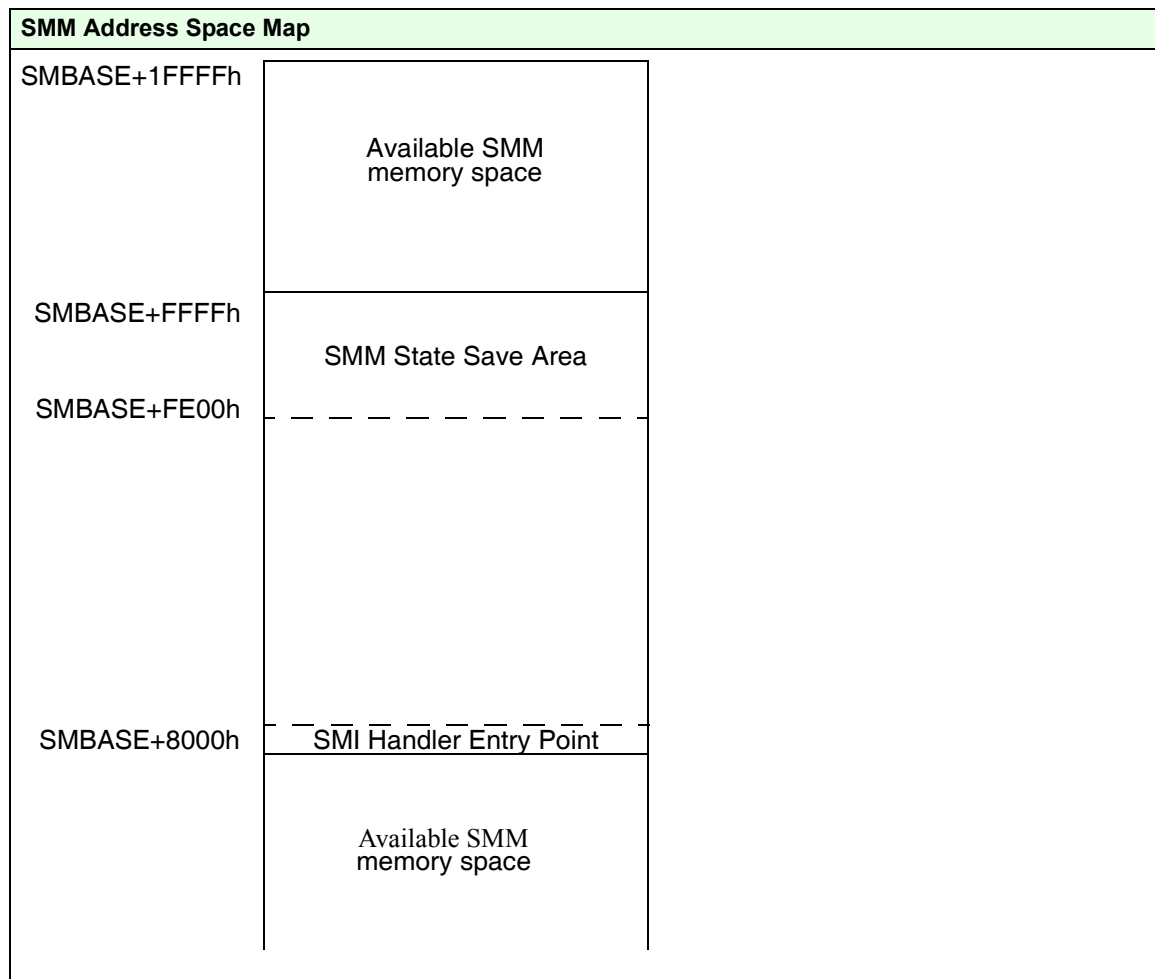
Similar to the standard SMM memory configuration, the high SMM memory configuration consists of two blocks of SMM memory: (1) extended SMM memory address space, and (2) high SMM memory address space. The extended SMM and high SMM memory together provide 1 MByte of continuous SMM memory located from (TOM) to (TOM + 1 MByte). Extended SMM memory ranges from (TOM) to (TOM + 896 KBytes) and high SMM memory ranges from (TOM) to (TOM + 1 MByte). The System Management RAM Control Register (SM_RAM_CR), and the Extended SMM RAM Control Register (ESM_RAM_CR) provide control bits for enabling/initializing extended SMM memory and high SMM memory.

5.1.3 Address Space Location and Use

The minimum size of the SMM memory space is 32 KBytes. The minimum address space starts at the SMM base address (SMBASE) plus an offset of 8000h and ends at the SMBASE plus an offset of FFFFh. SMBASE is 30000h initially.

The SMM address space may be relocated by modifying the SMBASE value in the SMM state save map from within an SMI service routine. The SMBASE value must always be aligned on a 32 KByte boundary. If

SMBASE is set to a non-aligned value in the state save map, a shutdown occurs during execution of the RSM instruction. The SMM address space map is shown in the following figure:



5.2 SMM Initialization and State Change

The recommended SMM initialization sequence is:

1. Program SM_RAM_CR and ESM_RAM_CR as desired to select the desired SMM memory configuration.
2. Open SMM memory space by setting the SM_OPEN bit in SM_RAM_CR to 1.
3. Transfer the normal SMI handler to the SMM address space.
4. Close SMM memory space by clearing the SM_OPEN bit.
5. Reprogram SMBASE by doing the following:

- a. Set up a temporary SMI handler at the default entry point (38000h). The purpose of the temporary SMI handler is to reprogram the SMM Base Address in the SMM state save area so that it aligns with the normal SMI handler.
 - b. Generate a test SMI using logic in the PCI-to-ISA bridge (southbridge) to execute the temporary SMI handler.
6. Generate a second test SMI to verify proper operation of the normal SMI handler.
 7. Optionally, set the SM_LOCK bit in the SM_RAM_CR register to 1 to lock down the SMM configuration.

5.2.1 State Save Map

When a system management interrupt (SMI) occurs, the processor saves its current operating state in the SMM state save area prior to executing the SMI handler.

The following table details the SMM state save map. The offset shown is relative to the SMI handler entry point (SMBASE+8000h). Some of the locations in the state save map are "reserved". Software should not modify or rely on the value of reserved locations.

SMM State Save Map

Offset	Description
7E00-7EF7h	Reserved
7EF8-7EFBh	SMM base address (SMBASE)
7EFC-7EFFh	SMM revision ID = 0003 0002h: Bit 17: SMBASE relocation supported Bit 16: I/O instruction restart supported Bits 15-0: SMM version 2
7F00-7F01h	I/O instruction restart Bits 15-0: 0000h = do not restart I/O 00FFh = restart I/O
7F02-7F03h	Auto HALT restart flag Bit 0 (read): 0 = processor not in HALT state 1 = processor in HALT state Bit 0 (write): 0 = do not resume to HALT state 1 = resume to HALT state
7F04-7F87h	Reserved
7F88-7F8Bh	GDTR base
7F8C-7F93h	Reserved
7F94-7F97h	IDTR base
7F98-7FC7h	Reserved
7FC8-7FCBh	DR7
7FCC-7FCFh	DR6
7FD0-7FD3h	EAX
7FD4-7FD7h	ECX
7FD8-7FDBh	EDX

SMM State Save Map (Continued)

Offset	Description
7FDC-7FDFh	EBX
7FE0-7FE3h	ESP
7FE4-7FE7h	EBP
7FE8-7FEBh	ESI
7FEC-7FEFh	EDI
7FF0-7FF3h	EIP
7FF4-7FF7h	EFLAGS
7FF8-7FFBh	CR3
7FFC-7FFFh	CR0

5.2.2 State Save Prior to Power Off

If software uses SMM for the purpose of implementing a power management state during which the power is removed from the processor, then the entire 512-byte SMM State Save Map (offset 7E00h-7FFFh) must be saved and restored for proper operation. Additionally, the following registers should also be saved and restored: CR2, DR0, DR1, DR2, DR3, FP and MMX registers, and MSRs.

5.2.3 Address Space Caching

SMM address space is always cached, and the TM5700/TM5900 makes sure the internal caches remain consistent. Therefore, the BIOS is not required to flush the internal caches on SMI entry or exit.

5.3 SMM Execution Environment

5.3.1 SMM Entry State

The processor enters SMM as the result of servicing an SMI. When an SMI is received, the processor first saves the current operating environment in the state save map area and then reprograms the processor's registers in preparation for servicing the SMI.

The following table lists the state of the registers that are reprogrammed when entering SMM. As shown, the PE and PG flags in CR0 are cleared when entering SMM. This places the processor in an operating mode similar to real address mode but with the following differences:

- Segment limits are 4 GBytes.

- The default operand and address sizes are set to 16 bits, as in real mode. This restricts the address space to 1 MByte. However, size override prefixes can be used to access the address space above 1 MByte.

SMM Entry State

Register	SMM Entry Value
CR0	EM, PE, PG, TS cleared, others unmodified
CR4	0000 0000h
CS selector	3000h
CS base	SMBASE
CS limit	FFFF FFFFh
CS access rights	009Bh, writeable ¹
DS, ES, FS, GS, SS selector	0000h
DS, ES, FS, GS, SS base	0000 0000h
DS, ES, FS, GS, SS limit	FFFF FFFFh
DS, ES, FS, GS, SS access rights	0093h, writable
DR7	0000 0400h
EFLAGS	0000 0002h
EIP	0000 8000h
TSC	Continues to increment

1. Whether or not CS is writable depends on the processor mode. CS is always writeable in real and virtual 8086 mode; CS is always unwriteable in protected mode.

5.3.2 Exceptions, Interrupts, and A20 Masking within SMM

When the processor enters SMM, all hardware interrupts are disabled. This includes SMI, NMI, INIT, INTR, single-step traps and breakpoint traps. If an SMI signal is asserted while servicing is disabled, then the new assertion will be serviced when the current SMI is finished being serviced. The new SMI occurrence is latched and serviced immediately after the processor executes the RSM instruction for the current SMI. If an NMI or INIT occurs while the processor is in SMM, one occurrence of the interrupt is latched until either an IRET or RSM instruction is executed. Once an IRET instruction has been executed within SMM, NMI and INIT requests are enabled and serviced immediately.

Upon entry to SMM, the state of A20M is changed to unmasked. The state of A20M may not be changed while operating in SMM. The CPU restores the original A20M state when the RSM instruction is executed. INTR, single step traps and breakpoints may be enabled from within SMM. However, an SMM interrupt table and necessary interrupt handlers must be created from within SMM to properly service these interrupts.

5.3.3 SMM Exit Considerations

The processor exits SMM following execution of a RSM instruction. The RSM instruction can only be executed while the processor is in SMM. If RSM is executed during normal operation, an invalid opcode exception is generated. The RSM instruction causes the processor to restore all values saved in the SMM state save map and then return control back to the interrupted program.

Consider the following conditions prior to executing the RSM instruction:

- SMBASE relocation
- interrupted HALT state
- interrupted I/O instruction

SMBASE Relocation

The SMBASE address may be relocated by modifying the SMBASE value in the SMM state save area prior to executing RSM. The SMBASE value must be aligned to a 32-KByte boundary. Attempting to load a misaligned value with the RSM instruction causes a processor shutdown.

Auto HALT Restart

If the processor was in a HALT state when the SMI occurred, the Auto HALT Restart flag in the SMM state save area will be set (bit 0, offset 7F02h). Prior to executing RSM, the SMI handler should check this flag and then implement one of the following if the flag is set:

- Leave the Auto Halt Restart flag set which causes the RSM instruction to return control to the interrupted HLT instruction. The processor then reissues a halt bus transaction and reenters the HALT state.
- Clear the Auto Halt Restart flag which causes the RSM instruction to return control to the instruction following the HLT instruction.

I/O Instruction Restart

If the SMI handler determines that the processor was executing an I/O instruction when the SMI occurred, the I/O instruction may be re-executed following the RSM instruction by using the I/O Instruction Restart flag (offset 7F00h) in the SMM state save area. If the I/O Instruction Restart flag is set to 00FFh, the RSM instruction returns control to the interrupted I/O instruction. If the I/O Instruction Restart flag is 0000h (default value), the RSM instruction returns control to the next instruction or, in the case of a repeated I/O instruction, to the next iteration of the interrupted instruction. The processor stores additional information in the SMM state save area to support the I/O instruction restart feature for the INS and OUTS instructions.

Power Management Configuration Code Example

The following code is an example of how to program the CPU's power management registers assuming a PIIX4-compatible southbridge is being used.

Power Management Configuration Code Example: PIIX4 Southbridge

```
/* PCI Configuration cycles */
#define CONF CYCLE (1L<<31) /* Configuration Cycle */
#define CONF ADDR 0x0cf8 /* Configuration Address register */
#define CONF DATA 0x0cfc /* Configuration Data register */

/* Southbridge power management I/O registers */
#define PMCNTRL 0x04 /* Power Management Control register */
#define P_LVL2 0x14 /* Processor Level 2 register */
#define P_LVL3 0x15 /* Processor Level 3 register */

/* Transmeta Northbridge: P_LVL2 monitoring */
#define P_LVL2_ADDR 0xE0 /* TM NB: PL2 monitor address */
#define P_LVL2_CTL 0xE6 /* TM NB: PL2 controls */
#define PL2RD 0x01 /* TM NB: PL2 monitor reads */
#define PL2WR 0x02 /* TM NB: PL2 monitor writes */

/* Transmeta Northbridge: P_LVL3 monitoring */
#define P_LVL3_ADDR 0xE8 /* TM NB: PL3 monitor address */
#define P_LVL3_CTL 0xEE /* TM NB: PL3 controls */
#define PL3RD 0x01 /* TM NB: PL3 monitor reads */
#define PL3WR 0x02 /* TM NB: PL3 monitor writes */

/* Transmeta Northbridge: STR monitoring */
#define STR_ADDR 0xF0 /* TM NB: STR monitor address */
#define STR_MASK 0xF2 /* TM NB: STR data mask */
#define STR_DATA 0xF4 /* TM NB: STR monitor data */
#define STR_CTL 0xF6 /* TM NB: STR controls */
#define STRRD 0x01 /* TM NB: STR monitor reads */
#define STRWR 0x02 /* TM NB: STR monitor writes */
```

Power Management Configuration Code Example: PIIX4 Southbridge (Continued)

```
/* Transmeta Northbridge: POS monitoring */
#define POS_ADDR      0xD8          /* TM NB: POS monitor address */
#define POS_MASK      0xDA          /* TM NB: POS data mask */
#define POS_DATA      0xDC          /* TM NB: POS monitor data */
#define POS_CTL       0xDE          /* TM NB: POS controls */
#define POSRD         0x01          /* TM NB: POS monitor reads */
#define POSWR         0x02          /* TM NB: POS monitor writes */

/* Helper: Set a PCI register (16-bit value) */
inline void
PCIregSet16(unsigned long reg, unsigned short val)
{
    outl(CONFADDR, CONF CYCLE + (reg & ~0x03));
    outw(CONFDATA + ((unsigned short)(reg & 0x03)), val);
}

/* Helper: Set a PCI register (8-bit value) */
inline void
PCIregSet8(unsigned long reg, unsigned char val)
{
    outl(CONFADDR, CONF CYCLE + (reg & ~0x03));
    outb(CONFDATA + ((unsigned short)(reg & 0x03)), val);
}

/* LVL2 configuration for PIIX4 */
PCIregSet16(P_LVL2_ADDR, <pmba> + P_LVL2);
PCIregSet8(P_LVL2_CTL, PL2RD);

/* LVL3 configuration for PIIX4 */
PCIregSet16(P_LVL3_ADDR, <pmba> + P_LVL3);
PCIregSet8(P_LVL3_CTL, PL3RD);

/* STR configuration for PIIX4 */
PCIregSet16(STR_ADDR, <pmba> + PMCNTL);
PCIregSet16(STR_MASK, 0x3C00);
PCIregSet16(STR_DATA, 0x2400);
PCIregSet8(STR_CTL, STRWR);

/* POS configuration for PIIX4 */
PCIregSet16(POS_ADDR, <pmba> + PMCNTL);
PCIregSet16(POS_MASK, 0x3C00);
PCIregSet16(POS_DATA, 0x3000);
PCIregSet8(POS_CTL, POSWR);
```

Power Management Configuration Code Example: ALI 1535 Southbridge

```

/* PCI Configuration cycles */
#define CONF CYCLE      (1L<<31)          /* Configuration Cycle */
#define CONF ADDR      0x0cf8            /* Configuration Address register */
#define CONF DATA     0x0cfc            /* Configuration Data register */

/* Southbridge power management I/O registers */
#define PMC CTRL      0x04              /* Power Management Control register */
#define P_LVL2       0x14              /* Processor Level 2 register */
#define P_LVL3       0x15              /* Processor Level 3 register */

/* Transmeta Northbridge: P_LVL2 monitoring */
#define P_LVL2_ADDR  0xE0              /* TM NB: PL2 monitor address */
#define P_LVL2_CTL   0xE6              /* TM NB: PL2 controls */
#define PL2RD        0x01              /* TM NB: PL2 monitor reads */
#define PL2WR        0x02              /* TM NB: PL2 monitor writes */

/* Transmeta Northbridge: P_LVL3 monitoring */
#define P_LVL3_ADDR  0xE8              /* TM NB: PL3 monitor address */
#define P_LVL3_CTL   0xEE              /* TM NB: PL3 controls */
#define PL3RD        0x01              /* TM NB: PL3 monitor reads */
#define PL3WR        0x02              /* TM NB: PL3 monitor writes */

/* Transmeta Northbridge: STR monitoring */
#define STR_ADDR     0xF0              /* TM NB: STR monitor address */
#define STR_MASK     0xF2              /* TM NB: STR data mask */
#define STR_DATA     0xF4              /* TM NB: STR monitor data */
#define STR_CTL      0xF6              /* TM NB: STR controls */
#define STRRD        0x01              /* TM NB: STR monitor reads */
#define STRWR        0x02              /* TM NB: STR monitor writes */

/* Transmeta Northbridge: POS monitoring */
#define POS_ADDR     0xD8              /* TM NB: POS monitor address */
#define POS_MASK     0xDA              /* TM NB: POS data mask */
#define POS_DATA     0xDC              /* TM NB: POS monitor data */
#define POS_CTL      0xDE              /* TM NB: POS controls */
#define POSRD        0x01              /* TM NB: POS monitor reads */
#define POSWR        0x02              /* TM NB: POS monitor writes */

/* Helper: Set a PCI register (16-bit value) */
inline void
PCIregSet16(unsigned long reg, unsigned short val)
{
    outl(CONF ADDR, CONF CYCLE + (reg & ~0x03));
    outw(CONF DATA + ((unsigned short)(reg & 0x03)), val);
}

```

Power Management Configuration Code Example: ALI 1535 Southbridge (Continued)

```
/* Helper: Set a PCI register (8-bit value) */
inline void
PCIregSet8(unsigned long reg, unsigned char val)
{
    outl(CONFADDR, CONFCYCLE + (reg & ~0x03));
    outb(CONFDATA + ((unsigned short)(reg & 0x03)), val);
}

/* LVL2 configuration for ALI 1535 */
PCIregSet16(P_LVL2_ADDR, <pmba> + P_LVL2);
PCIregSet8(P_LVL2_CTL, PL2RD);

/* LVL3 configuration for ALI 1535 */
PCIregSet16(P_LVL3_ADDR, <pmba> + P_LVL3);
PCIregSet8(P_LVL3_CTL, PL3RD);

/* STR configuration for ALI 1535 */
PCIregSet16(STR_ADDR, <pmba> + PMCNTL);
PCIregSet16(STR_MASK, 0x3C00);
PCIregSet16(STR_DATA, 0x2C00);
PCIregSet8(STR_CTL, STRWR);

/* POS configuration for ALI 1535 */
PCIregSet16(POS_ADDR, <pmba> + PMCNTL);
PCIregSet16(POS_MASK, 0x3C00);
PCIregSet16(POS_DATA, 0x2400);
PCIregSet8(POS_CTL, POSWR);
```


PCI Passive Release

When using some chipset PCI-to-ISA bridge (southbridge) devices with the Crusoe™ Processor Model TM5700/TM5900, the chipset Passive Release functions should be disabled. Disable this function as

appropriate for each particular PCI-to-ISA bridge. The chipsets listed below must have these this function disabled.

Note
Some peripheral devices, sound cards in particular, are known to re-enable the Passive Release function. See the release notes for a list of peripheral cards for which this is known to occur.

Example: Disabling Intel PIIX4 Passive Release

The PIIX4 Passive Release function is disabled by clearing bits 1 and 2 of the Deterministic Latency Control (DLC) register. The DLC register is a PCI Configuration register of the PIIX4 PCI-to-ISA bridge (function 0, offset 82h).

Example: Disabling ALI 1535 Passive Release

The ALI 1535 Passive Release function is disabled by clearing bit 4 of the PCI Interface Control (PIC) register. The PIC register is a PCI Configuration register of the ALI 1535 PCI-to-ISA bridge (function 0, offset 40h).

Initial Processor State

The following table documents the initial processor state after RESET and after INIT.

During RESET, the processor is fully initialized, so that execution begins from a known state. During INIT, most of the processor is initialized. However, certain registers and units remain unchanged to allow a faster restart process. No separate shutdown state exists. If a shutdown occurs, the processor reacts the same way it reacts to the assertion of INIT: to cause a restart.

Table 17: Initial Processor State

Register	Value after RESET	Value after INIT
EAX	0000 0000h	0000 0000h
EBX	0000 0000h	0000 0000h
ECX	0000 0000h	0000 0000h
EDX	0000 0543h	0000: CPUID_TFMS MSR
ESI	0000 0000h	0000 0000h
EDI	0000 0000h	0000 0000h
EBP	0000 0000h	0000 0000h
ESP	0000 0000h	0000 0000h
EIP	0000 FFF0h	0000 FFF0h
EFLAGS	0000 0002h	0000 0002h
CS	selector = F000h base = FFFF 0000h limit = 0000 FFFFh access rights = 009Bh ¹	selector = F000h base = FFFF 0000h limit = 0000 FFFFh access rights = 009Bh ¹
SS, DS, ES, FS, GS	selector = 0000h base = 0000 0000h limit = 0000 FFFFh access rights = 0093h	selector = 0000h base = 0000 0000h limit = 0000 FFFFh access rights = 0093h
GDTR, IDTR	base = 0000 0000h limit = 0000 FFFFh	base = 0000 0000h limit = 0000 FFFFh
LDTR, TR	selector = 0000h base = 0000 0000h limit = 0000 FFFFh access rights = 0082h	selector = 0000h base = 0000 0000h limit = 0000 FFFFh access rights = 0082h
CR0	6000 0010h	x000 0010h ²
CR2	0000 0000h	0000 0000h

Table 17: Initial Processor State

Register	Value after RESET	Value after INIT
CR3	0000 0000h	0000 0000h
CR4	0000 0000h	0000 0000h
DR0	0000 0000h	0000 0000h
DR1	0000 0000h	0000 0000h
DR2	0000 0000h	0000 0000h
DR3	0000 0000h	0000 0000h
DR6	FFFF 0FF0h	FFFF 0FF0h
DR7	0000 0400h	0000 0400h
ST0-ST7	+0.0	unmodified
MM0-MM7	0000 0000 0000 0000h	unmodified
CW	0040h	unmodified
SW	0000h	unmodified
TW	5555h	unmodified
FP_IP	0000: 0000 0000h	unmodified
FP_DP	0000: 0000 0000h	unmodified
FP_OPC	000 0000 0000b	unmodified
TSC MSR	0000 0000 0000 0000h	unmodified
PSN_DISABLE MSR	determined by configuration	unmodified
CPUID_TFMS MSR	0543h	unmodified
CPUID_VND1 MSR	756E 6547h	unmodified
CPUID_VND2 MSR	5465 6E69h	unmodified
CPUID_VND3 MSR	3638 784Dh	unmodified
CPUID_MASK MSR	FFFF FEFFh	unmodified
LONGRUN MSR	0000 0064 0000 0064	unmodified
A20 Masking	unmasked	unmodified
Processor caches	invalidated	unmodified
TLBs	invalidated	invalidated

1. Whether or not CS is writable depends on the processor mode. CS is always writable in real and virtual 8086 mode; CS is always unwriteable in protected mode.
2. Bits 29 (NW) and 30 (CD) remain unmodified.

BIOS and Keyboard Controller Checklists

This document provides checklists for assisting BIOS and keyboard controller developers assure that Crusoe processor-specific issues are addressed appropriately during development and bring-up. BIOS and keyboard controller development checklists are provided in the tables below.

BIOS Bring-up Checklist

Item	BIOS / KBC Bring-up Tasks	Status
1.	Bring-up BIOS for first platform	
2.	PCI IRQ routine implement support	
3.	GPIO initial support	
4.	Memory configuration (autosizing/autotyping)	
5.	System power-on	
6.	Video initialization	
7.	Video display (CRT)	
8.	Video display (LCD)	
9.	Keyboard operation (internal/external)	
10.	FDD read/write	
11.	HDD read/write	
12.	L2 cache auto-sizing and initialization	
13.	Track-point/pad function support	
14.	External PS2 function support	
15.	Serial port	
16.	Parallel port	
17.	Boot to MS-DOS	
18.	Boot to Windows	

Standard Implementation Checklist

Item	BIOS / KBC Standard Implementation Tasks	Status
1.	PCMCIA initialization	
2.	PS2 mouse hot-plug	
3.	Audio function support	

Standard Implementation Checklist

Item	BIOS / KBC Standard Implementation Tasks	Status
4.	CPU type detection	
5.	Ultra DMA function	
6.	Keyboard download matrix	
7.	BIOS boot block support	
8.	SMI initialization	
9.	APM 1.2	
10.	Fast A20 for HIMEM.SYS	
11.	Flash ROM function support (utility)	
12.	Parallel (EPP/ECP modes)	
13.	BIOS INT 13 extension	
14.	SM BIOS 2.1 / DMI 2.0	
15.	HDD/CD auto typing	
16.	HDD auto detection	
17.	3 mode FDD support	
18.	Multi-boot (FDD, HDD, CD)	
19.	Read / write / format LS-120	
20.	Cardbus function	
21.	FIR/SIR support	
22.	Fax/modem support	

Customization Checklist

Item	BIOS / KBC Customization Tasks	Status
1.	Quick boot	
2.	Quiet boot	
3.	Security / password	
4.	USB initialization	
5.	SM bus support	
6.	Ultra DMA	
7.	Battery gauge function	
8.	Extension INT 15 support	
9.	Dual function power switch	

Plug-n-Play Checklist

Item	BIOS / KBC Plug-n-Play Tasks	Status
1.	ESCD	
2.	CDR	
3.	SIO MCD	
4.	Audio MCD	
5.	Video MCD	

Plug-n-Play Checklist

Item	BIOS / KBC Plug-n-Play Tasks	Status
6.	LPT MCD	
7.	COM MCD	
8.	KBC MCD	

Power Management Checklist

Item	BIOS / KBC Power Management Tasks	Status
1.	DOZE mode	
2.	Standby mode	
3.	Save-to-RAM (STR)	
4.	Resume from STR	
5.	Save-to-Disk (STD)	
6.	Resume from STD	
7.	HDD timeout	
8.	Video timeout	
9.	FDD PMSR	
10.	HDD PMSR	
11.	CD-ROM PMSR	
12.	Audio PMSR	
13.	Video PMSR	
14.	SIO PMSR	
15.	CardBus PMSR	
16.	KBC PMSR	
17.	Battery low action	
18.	Battery critical low action	
19.	Cover door close / open	
20.	Clock throttling	
21.	Thermal management / CoolRun	
22.	AC in / out SMI	
23.	Battery in / out SMI	
24.	Modem ring wake up from STR	
25.	RTC alarm wake up from STR	
26.	Modem ring wake up from STD	
27.	RTC alarm wake up from STD	
28.	Wake-on-LAN from STD	
29.	Wake-on-LAN from STR	
30.	AC in / out beep	
31.	System activity monitor check	
32.	VGA activity monitor check	

Hotkey Checklist

Item	BIOS / KBC Hotkey Tasks	Status
1.	<Fn> <F2> fuel gauge utility open / close	
2.	<Fn><F7> select display	
3.	<Fn><F4> Save-to-RAM	
4.	<Fn><F12> Save-to-Disk	
5.	<Fn><F3> system standby	
6.	<Fn><F8> LCD expand	
7.	<Fn><F11> max performance	

Setup Checklist

Item	BIOS / KBC Setup Tasks	Status
1.	Audio features - setup	
2.	Security menu - setup	
3.	Power savings menu - setup	
4.	Advanced menu - setup	
5.	Exit menu - setup	
6.	Video features - setup	
7.	Boot sequence - setup	
8.	Integrated peripherals - setup	
9.	Keyboard / mouse - setup	

ACPI Checklist

Item	BIOS / KBC ACPI Tasks	Status
1.	Install ACPI and bring up Memphis	
2.	PCI IRQ routing ASL	
3.	_CRS for PCI0 (ACPI)	
4.	ACPI SIO configuration support w/ integration with BIOS setup	
5.	ACPI audio configuration support w/ integration with BIOS	
6.	ACPI - VGA, PC card, other devices	
7.	ACPI - rest of devices	
8.	ACPI device power management for applicable devices (D0-D3 states)	
9.	ACPI - S1	
10.	ACPI - S2	
11.	ACPI - S3	
12.	ACPI - S4 BIOS	
13.	ACPI - S5	
14.	ACPI - battery - SMB	
15.	ACPI - ASL for wake events and EC query	
16.	ACPI - ASL for PME events	
17.	ACPI - ASL for docking	

ACPI Checklist

Item	BIOS / KBC ACPI Tasks	Status
18.	ACPI - ASL for device bay	
19.	ACPI - ASL and SMI code for hot-keys (list individually)	
20.	ACPI interface to thermal sensor	
21.	ACPI - ASL for thermal management (active and passive) / CoolRun	
22.	ACPI - ASL for fan control	

Docking Checklist

Item	BIOS / KBC Docking Tasks	Status
1.	Cold docking	
2.	PCI-to-PCI bridge support	
3.	Dock in (warm / hot)	
4.	Dock out (warm / hot)	
5.	CardBus support	

Miscellaneous Checklist

Item	BIOS / KBC Miscellaneous Tasks	Status
1.	Flash utility for all BIOS ROM	
2.	CMS upgrade utility	
3.	PHDISK utility	
4.	WHQL	

Note

Set bit 3* in the MISC_CNTL2 (0xa7) register of the PMU device of the ALI 1535 southbridge. This allows the southbridge to assert—and then immediately deassert—STPCLK in the case that there is already a pending break event when the STPCLK is requested.

If this bit is not set, the STPCLK is postponed until the break event has been cleared. This can lead to a situation where the STPCLK is delivered at a time when no possible break events can occur and so the system never wakes up.

This issue is specific to the ALI 1535. Note that, although the issue is visible with Windows XP, it applies to any ACPI-aware operating system.

* MISC_CNTL2 Bit 3: enable clock control, don't check break event in advance⁷

