Using the Intel® ICH Family Watchdog Timer (WDT)

Application Note: AP-725

September 2002

Revision 001

Document Number: 292273-001

Contents

Tables

Revision History

Rev. No.	Description	Rev. Date
1.0	Initial Release.	September 17, 2002

This page is intentionally left blank.

1 Background

1.1 Purpose of this Document

This document describes the Watchdog Timer functions in the Intel® ICH products family and provides additional details on how the Watchdog Timer function can be used and controlled by software. This document also describes different options available for the software in its interaction with the hardware.

Note: In this document the acronym TCO (Total Cost of Ownership) refer to a logic block in the Intel ICH products family. The Watchdog Timer (WDT) is one of the functions of that logic block.

1.2 Watchdog Timer Objectives

There are many reasons why a computer might cease to operate correctly. A few of the possible reasons include:

- A software error can cause the Operating System (OS) to stop scheduling tasks.
- A software error may leave certain interrupts disabled or masked, preventing other software from continuing until an interrupt is completed.
- A hardware error may prevent data from flowing properly within the system.
- A hardware error can halt operation of the CPU.

A key assumption is required: a correctly functioning OS can schedule tasks to be run on a periodic basis. If a task is not able to be completed within a predetermined bounded time, a separate monitoring circuit can conclude that the system is malfunctioning. The monitoring circuit, often called a Watchdog Timer (WDT), must be able to independently track the completion of a task within its time limit. It must be able to do so independently of the OS's state, the CPU's state or ability to run the OS, or the state of any other hardware device that might lead to a system malfunction.

Upon determining that its time limit has been reached, the WDT circuit concludes that an error has occurred and may take a variety of corrective actions. For example, it may attempt to generate a non-maskable interrupt, reset the platform, or even force the shut-down of the platform. The WDT may react in different ways depending on the other factors, such as measured activity from the CPU or reception of interrupts from peripherals.

Working in conjunction with an appropriate BIOS and OS, the WDT enables the platform to detect and take corrective action from a variety of errors. The function is intended to meet the following objectives:

- Detect and recover from either hardware or software errors that lock up the platform.
- Allow a programmable amount of time to pass before concluding that the hardware has malfunctioned.
- Support a programmable response to the lockup, including resetting the platform or shutting it down.
- Minimize software overhead required to manage the WDT.
- Provide information for recovery software to determine why the platform was reset (i.e. reset was due to WDT expiration).

Long term, the platform should be able to detect and recover from lockups that occur during the BIOS to OS handoff. The software mechanisms required to enable this have not been defined as of this revision of this document. It is anticipated that a future revision may document these mechanisms.

1.3 Hardware Overview

The ICH's TCO logic is comprised of several key elements:

Table 1: TCO Logic Key Elements

Element	Description
Countdown Timer	Software programs the Watchdog Timer (WDT) with an initial value. The WDT then counts down to zero. When it reaches zero, it signals to the other elements that it has reached zero. The WDT can also be reloaded automatically by the Reaction logic.
Reaction	This logic takes the various actions based on the timeout of the WDT and the various Configuration bits. It also has the capability to reload the WDT.
Configuration Bits	These bits select the operating modes of the logic.
Status Bits	These bits report the state of the WDT logic.
Mail Boxes	This logic allows communication between the BIOS and OS. The BIOS can generate interrupts to the OS and the OS can call the BIOS by causing an SMI.

NOTE:
Throughout this document, the term BIOS will be used to refer to the platform firmware.

2 Functional Definition

2.1 Mapping of Registers, Bits, and Straps

Most of the registers and bits associated with the TCO logic are mapped into the system I/O space. The base address for this I/O space is the same as used for the ICH's ACPI I/O space. This is programmed through the PCI config space, Device 31, Function 0, Offset 40h. This allows the I/O space to be located anywhere in the CPU's 64K I/O space. In addition to the I/O space, there is a related bit in the PCI config space and a pin strap.

Note: The TCOBASE address is always ACPIBASE + 60h in the PCI config space.

The registers, bits, and pin straps are described below.

Offset	Mnemonic	Register Name: Function
00h	TCO_RLD	WDT Reload and Current Value: Used to reload the WDT and read its current value.
01h	TCO_TMR	WDT Timer Initial Value: Used to set the value that is reloaded in the WDT when the TCO_RLD register is used.
02h	TCO_DAT_IN	TCO Data In: Used by the OS to send messages to the BIOS.
03h	TCO_DAT_OUT	TCO Data Out: Used for the BIOS to send message to the OS.
04h–05h	TCO1_STS	TCO Status: Used to read the status of the TCO logic
06h–07h	TCO2_STS	TCO Status: Also used to read the status of the TCO logic
08h–09h	TCO1_CNT	TCO Control: Used to control the TCO logic

2.2 Detailed Register, Bit, and Pin Descriptions

2.2.1 TCO_RLD – TCO Timer Reload and Current Value Register

Offset:	TCOBase + 00h	Attribute:	Read/Write
Default Value:	00h	Size:	8 bits

Bit	Description
7:6	Software associated must ignore these bits when read and always write 0 to these bit positions.
5:0	**TCO Timer Value**. Reading this register will return the current count of the WDT. Writing any value to this register will automatically reload the timer with the value last written to the TCO_TMR register. This will prevent the WDT from reaching zero. **NOTE:** The value reported when reading this register may be ambiguous because software cannot determine if the WDT has already reached 0 and has been reloaded.

2.2.2 TCO_TMR – TCO Timer Initial Value Register

Offset:	TCOBase + 01h	Attribute:	Read/Write
Default Value:	04h	Size:	8 bits

Bit	Description
7:6	Software associated must ignore these bits when read and always write 0 to these bit positions.
5:0	**TCO Timer Initial Value**. Value that is loaded into the WDT each time the TCO_RLD register is written. **NOTES:** 1. Values of 00h or 01h will be ignored and should not be attempted (i.e. software must not write these values to this register). 2. The default value of 04h results in the first timeout 1.8 to 2.4 seconds after system reset goes inactive. The second timeout will occur at 4.2 to 4.8 seconds after system reset goes inactive.

2.2.3 TCO_DAT_IN – TCO Data In Register

Offset:	TCOBase + 02h	Attribute:	Read/Write
Default Value:	00h	Size:	8 bits

Bit	Description
7:0	**TCO Data In Value**. The OS may write to this register to pass values to the BIOS. When the OS writes to this register, the hardware will automatically cause an SMI. **NOTE:** The values written to this register are just a convention between the OS and BIOS. The conventions associated with this mailbox have not yet been defined.

NOTE: This is also known as the OS-to-BIOS Mailbox. It has no impact on the WDT.

2.2.4 TCO_DAT_OUT – TCO DATA Out Register

Offset:	TCOBase + 03h	Attribute:	Read/Write
Default Value:	00h	Size:	8 bits

Bit	Description
7:0	**TCO Data Out Value**. The BIOS may write to this register to pass values to the OS. When the BIOS writes to this register, the hardware will automatically cause an interrupt. The selection of the interrupt is configured by the BIOS using a register that is beyond the scope of this document. The interrupt is cleared by writing a 1 to the TCO_INT_STS bit. **NOTE:** The values written to this register are just a convention between the OS and BIOS. The conventions associated with this mailbox have not yet been defined.

NOTE: This is also known as the BIOS-to-OS Mailbox. It has no impact on the WDT.

2.2.5 TCO1_STS – TCO1 Status Register

Offset:	TCOBase + 04h	Attribute:	Read/Write
Default Value:	00h	Size:	16 bits

Bit	Description
15:4	Reserved
3	**Time Out Status (TIMEOUT)**. This bit will be set to 1 by the hardware to indicate that the WDT has reached 0. This bit remains set until cleared by software writing a 1 to this bit position. **NOTE:** When the TCO_EN bit is set in the SMI_Control and Enable register (PMBASE + 30h) to enable the TCO SMI, the SMI handler must clear the TIMEOUT bit, or repeated and immediate re-entry to the SMI handler (aka SMI storm) will result. Clearing the TIMEOUT bit will result in a TCO SMI being generated when the WDT timer next times out, not a reset.
2	**TCO Interrupt Status (TCO_INT_STS)**. This bit will be set to 1 by the hardware to indicate that the interrupt was caused by the write to the TCO_DAT_OUT register (BIOS-To-OS Mailbox). This bit remains set until cleared by the software writing a 1 to this bit position.
1	**Software TCO SMI Status (SW_TCO_SMI)**. This bit will be set to 1 by the hardware to indicate that the SMI was caused by the write to the TCO_DAT_IN register (OS-To-BIOS Mailbox). This bit remains set until cleared by the software writing a 1 to this bit position.
0	Reserved

NOTE: Bits marked “Reserved” in this register are either reserved or associated with functions other than the TCO logic. Software associated with the WDT function must ignore these bits when read and always write 0 to these bit positions.

2.2.6 TCO2_STS – TCO2 Status Register

Offset:	TCOBase + 06h	Attribute:	Read/Write
Default Value:	0000h	Size:	16 bits

Bit	Description
15:3	Reserved
2	**Boot Status (BOOT_STS)**. This bit will be set to 1 when the WDT reaches 0 after reset and the CPU fails to fetch an instruction. This is an indication that the CPU failed to come out of reset.
1	**Second TCO Time-Out Status (SECOND_TO_STS)**. This bit will be set to 1 when the WDT reaches zero two consecutive times. This is generally the result of a hardware error, since the SMI handler was unable to reload the WDT. This bit is only cleared by writing a 1 to this bit. **NOTE:** Software must clear the SECOND_TO_STS bit first, then the BOOT_STS bit (using 2 separate I/O write operations)
0	Reserved

NOTE: Bits marked "Reserved" in this register are either reserved or associated with functions other than the WDT logic. Software associated with the WDT function must ignore these bits when read and always write 0 to these bit positions.

2.2.7 TCO1_CNT – TCO1 Control Register

Offset:	TCOBase + 08h	Attribute:	Read/Write
Default Value:	0000h	Size:	16 bits

Bit	Description
15:12	Reserved
11	**TCO Timer Halt (TCO_TMR_HALT)** 1 = The WDT will halt. It will not count, and thus cannot reach a value that would cause a timeout. 0 = The WDT is enabled to count (default). Note that the WDT will not reset or reload when this bit is changed. If the WDT is halted and then enabled, the WDT will continue counting starting at the value where it was halted.
10	Reserved
9	Software must always write back to this bit the same value read. ***Warning:*** This bit is not associated with the TCO function. If software needs to modify the TCO_TMR_HALT bit, it must read the entire register and write back the updated value for bit 11 (TCO_TMR_HALT) and write back with the value read for bit 9.
8:0	Reserved

NOTE: Bits marked "Reserved" in this register are either reserved or associated with functions other than the TCO logic. Software associated with the WDT function must ignore these bits when read and always write 0 to these bit positions.

2.2.8 NO_Reboot Config Bit

Location:	D31:F0, Offset D4h	Attribute:	Read/Write
Default Value:	0000h	Size:	8 bits

Bit	Description
8:2	Reserved
1	**NO_REBOOT –** R/W (special). Hardware sets this bit to 1 if the ICH has been strapped to the “No Reboot” configuration. This bit may also be written to 1 or 0 by software if the strap indicates reboot. However, software cannot override the strap when it indicates “No Reboot” (if the strap forces this bit to 1, software cannot write it to 0). If NO_REBOOT = 1 (by strap or software), the WDT will count down and generate the SMI after the first timeout. However, if it reaches a 2nd timeout, ICH will not reboot the platform. If NO_REBOOT = 0, the ICH will reboot the system after the second timeout. The reboot is done by the ICH asserting PCIRST#. **NOTE:** The strap can be used to prevent a reboot if an ITP is used. It should also be used for servers that will have a BIOS that takes more than 4 seconds during boot before it can reach a point where it can reload the TCO.
0	Reserved

NOTE: This register is in PCI config space.

NOTE: Bits marked "Reserved" in this register are either reserved or associated with functions other than the TCO Timer logic. Software associated with the TCO Timer function must ignore these bits when read and always write 0 to these bit positions.

2.3 Basic Functional Description

- Once loaded, the WDT will decrement approximately every 0.6 seconds until it reaches zero. The timer ticks are accurate to within the same accuracy as the system's PCI clock, but there is always a one-tick uncertainty (the first tick after a reload may be up to 0.6 seconds after the reload).
- The decrement period is also subject to small variations, and may not be exactly 0.6 seconds. For example, the crystal that drives the clock may have drift, there may be deliberate variations in the frequency due to spread-spectrum techniques, or there may be deliberate over-clocking of the platform, etc. All timing values should be taken as approximate.
- The WDT can be loaded with a valid value at any time.
- The WDT value may not be written with a value of 00h or 01h for ICH1 through ICH4 or 00h to 03h for ICH5, as this will cause functional failures.
- The maximum value of 3Fh (decimal 63) will result in a timeout of approximately 37.5 seconds. The SMI handler may be used to extend the timeout period.
- When the WDT reaches zero, the ICH takes several steps:
    - Automatically reloads and starts the WDT counting toward zero
    - The TIMEOUT bit is set to indicate the first timeout
    - SMI is generated (optional if enabled by BIOS)
- If the SMI is generated, the SMI handler is then given an opportunity to determine why the WDT reached zero.
- If the WDT reaches zero a second time while the TIMEOUT bit is set (i.e. the SMI handler wasn't able to run and reload the timer or clear the TIMEOUT bit), then the WDT logic will cause the ICH to reset the platform.
- After resetting the platform, if the WDT times out a third time, then the platform is considered unbootable and no further attempts will be made to reset.
- The default value for the WDT (04h) is approximately 1.8 to 2.4 seconds, so it takes 1.8 to 2.4 seconds for the first timeout. The second timeout will occur 2.4 seconds later (4.2 to 4.8 seconds after system reset goes inactive). The BIOS is assumed to activate within 4.2 seconds and either disable or reset the WDT. If a system is unable to activate the BIOS within 4.2 seconds after the power is valid, such as when an ITP is used, then a hardware strap must be used to prevent the WDT logic from rebooting the platform.
- The WDT only counts down in an S0 state.
    - If the platform goes to an S1 state, the WDT will stop counting. It will automatically restart when the system returns to an S0 State.

  - When the system goes to an S3, S4, or S5 state, the WDT is unpowered. It will be reset when the platform returns to S0, and must be reinitialized by software.

- The platform is assumed to go to a sleeping state when the SLP_EN bit is set in the ACPI register space. If the platform is brought to an S5 state using some other method, such as a power button override, the WDT will stop counting and will not be able to reset the platform.

- If the SECOND_TO_STS bit is set, but the BOOT_STS is not set, the BIOS can conclude that the system rebooted due to some system hardware lockup, but not due to a CPU booting issue.

2.4 Detecting an Unbootable CPU or System

A CPU may fail to reset if it has been inserted incorrectly, is somehow damaged (i.e., due to vibrational stress), if the chipset itself is not working properly, or if the CPU is missing or not responding at the time the chipset attempts the reset. After the chipset attempts to reset the CPU, the CPU is expected to fetch its first instruction.

The TCO logic can detect this type of failure because the WDT will timeout three consecutive times: the first possibly causing an SMI, the second causing a reset, and the third leading to the conclusion that the CPU is not responding. At this point there is no reason to reset the platform, so the TCO logic sets the BOOT_STS bit and shuts down the platform.

The chipset may also have the capability to report the BOOT_STS bit through various external connections, such as the SMBus interface. This is available to an external microcontroller or LAN controller to report to other devices on the network.

If the system is later brought back to an S0 state, the BIOS will see that the BOOT_STS bit is set and know that the system had shut down due to a non-responsive CPU. The BIOS can clear the BOOT_STS bit by writing a 1 to that bit position.

No specific software is required in order to detect an unbootable CPU or system.

This page is intentionally left blank.

3 Algorithms and Software Requirements

3.1 Overview of the Watchdog Timer

The WDT is used to detect lockups that occur during the operation of the BIOS/OS and does not require any interaction between the BIOS and OS. It may or may not detect lockups that occur during the hand-off when the BIOS loads the OS, or during some part of the wake sequence from S3, S4, or S5 state. It also may not provide all details to the OS indicating the reasons for the reboot after a lockup.

There are many ways in which the software could interact with the hardware. However, the method described in this specification uses an ACPI description table to indicate the presence of a WDT that meets a well-defined set of requirements. See Section 4 for details on the ACPI WDT Descriptor Table.

3.2 Initializing the Watchdog Timer

The WDT logic requires some initialization by software.

When the platform comes out of reset, during a cold boot or wake from low-power states such as S3-S5, the WDT is running. This is needed to handle the cases where there are problems during the boot. If the BIOS is not going to use the WDT beyond coming out of reset, it can just halt the timer (to prevent it from reaching 0).

At some point the BIOS or OS may choose to enable the WDT. In order to use it, the BIOS or OS must initialize the WDT logic. This involves the following steps:

1. Set the timeout value by writing to the TCO_TMR register.
2. Clear status bits in the TCO1_STS and TCO2_STS registers.
3. Enable reboots (if needed) by clearing the No_Reboot bit if it is set.
4. Force the timer to its reload value by writing to the TCO_RLD register.
5. Enable the timer by clearing the TCO_TMR_HLT bit in the TCO1_CNT register

3.3 General Algorithms

The general algorithm is:

1. The BIOS uses the WDT to detect lockups that occur during the BIOS boot phase. It does so by periodically reloading the timer to keep it from reaching 0.
2. If the WDT does reach 0, the logic will cause SMI to go active. The BIOS should not enable the WDT to cause an SMI until the SMI handler has been loaded.
3. Before the BIOS starts loading the OS, it should complete the ACPI WDT Descriptor Table and halt the WDT by setting the TCO_HALT bit.

The BIOS then starts the OS loader. Once it is loaded, the OS (or the OS loader) may start using the WDT.

1. The OS or OS loader looks for the presence of the ACPI WDT Descriptor Table to indicate the presence, location and status of the WDT.
2. The OS should program the desired values in the TCO_TMR register. For example, this might be 30 seconds.
3. The OS should write to the TCO_RLD register to cause the WDT to be reloaded.
4. The OS should clear the TCO_HALT bit to start the WDT.
5. The OS must periodically reload the WDT using the TCO_RLD register to prevent the timer from reaching 0. The rate at which the WDT is reloaded must be faster than the timeout.

If the OS or hardware locks up and allows the timer to reach 0, the WDT will optionally generate an SMI (if the BIOS has enabled it). At a minimum, the SMI handler is expected to set the TCO_TMR register to its minimum allowable value, write to the TCO_RLD register and then stall. This will cause the WDT to reset the system. The SMI handler can be used to extend the timeout period or perform other actions based on the apparent system unresponsiveness.

After the reset, the BIOS will see the SECOND_TO_STS bit set. This informs the BIOS that the platform is coming out of a reset initiated by the WDT logic. The BIOS should copy this bit to the ACPI WDT Descriptor Table's TCO Event bit in the Status field and then clear the hardware bit. The OS will use the WDT Event bit in the WDT Descriptor Table to determine the reason for the last system reset.

3.4 Handling Sleep States

The OS and BIOS have roles and responsibilities related to transitions into and out of the S1, S3, S4 and S5 sleep states.

Before telling the platform to enter an S1-S5 state, it is also recommended that the OS reload the WDT by writing to the TCO_RLD register, and then halt the WDT.

When returning from the sleeping state, if the BIOS runs before the OS starts, it should initialize the WDT. It may optionally use the WDT to monitor its own progress. The BIOS should stop the WDT immediately before handing control back to the OS. The OS should then continue at step 2 in the sequence above.

Other algorithms are possible if it is desired to cover the BIOS to OS handoff. These may be documented in a future revision of this document.

3.5 Handling Resets

The BIOS has no responsibility to handle a WDT event except to set the WDT Event bit in the ACPI WDT Descriptor Table. However, there is a wide range of actions it might take, such as logging the event, running diagnostics, sending an Alert On LAN (AOL) message or even booting from an alternate disk partition. Details of these actions are beyond the scope of this document.

This page is intentionally left blank.

4 ACPI WDT Descriptor Table

The platform uses an ACPI description table to pass information about the WDT implementation to the operating system. This table contains information that allows the operating system to identify the specific WDT hardware as well as where it is located. The table also passes other information to the OS such as whether the user has disabled the WDT in BIOS setup, if the system was reset as a result of a WDT event, etc. The table will be abbreviated as WDDT.

The OS, OS loader or a device driver can detect the presence of a WDT by looking for the 'WDDT' signature in memory. See the ACPI 2.0 specification for further information.

4.1 ACPI WDT Descriptor Table Format

Field	Byte Length	Byte Offset	Description
Table Header			
Signature	4	0	'WDDT' is the signature for the ICH Watchdog Timer Descriptor Table.
Length	4	4	Length, in bytes, of entire WDDT.
Revision	1	8	1
Checksum	1	9	Entire table must sum to zero.
OEMID	6	10	OEM ID.
OEM Table ID	8	16	For the WDDT, the table ID is the manufacturer model ID.
OEM Revision	4	24	OEM revision of the WDDT for the supplied OEM Table ID.
Creator ID	4	28	Vendor ID of the utility that created the table. For the WDDT, this is the ID for the ASL Compiler.
Creator Revision	4	32	Revision of the utility that created the table. For the WDDT, this is the revision for the ASL Compiler.
Table Contents			
TCO Spec Version	2	36	Intel Watchdog Timer Specification Version 1.0 = 0x0100
TCO Description Table Version	2	38	Version of this table. Version 1.0 = 0x0100
PCI Vendor ID	2	40	Must be 0xFFFF if it is not a PCI device.
TCO Base Address	12	42	The base address of the WDT described using the Generic Address Structure.

Field	Byte Length	Byte Offset	Description
Timer Max Count	2	54	Contains the maximum counter value that this WDT implementation supports.
Timer Min Count	2	56	Contains the minimum allowable counter value that this WDT implementation supports.
Timer Count Period	2	58	Contains the period of one count specified in milliseconds that this WDT supports. For example a WDT counting in seconds would report 1000.
Status	2	60	Bit 0 – WDT Available bit (BIOS setup) 0 = permanently disabled 1 = available Bit 1 – WDT Active bit (BIOS Setup). **NOTE:** This bit is independent of the Ownership bit. 0 = WDT stopped when BIOS hands off control 1 = WDT running when BIOS hands off control Bit 2 – reserved for Ownership bit. **NOTE:** This bit indicates the ownership of the WDT when the BIOS Handoff Support is set in the capability field. 0 = TCO is owned by they BIOS 1 = TCO is owned by the OS Bits 3-10 – Reserved Bit 11 – User Reset Event bit 1 = system reset result of a user-initiated reset Bit 12 – WDT Event bit 1 = system reset result of WDT event Bit 13 – Power Fail Event bit 1 = system reset result of abnormal power event Bit 14 – Unknown Reset Event bit 1 = system reset result of a reset that was non user initiated and not due to the WDT logic Bit 15 – Reserved
Capability	2	62	Bit 0 – Auto Reset Bit 1 – Alert Support Bit 2 – Reserved for Platform Directed Shutdown Bit 3 – Reserved for Immediate Shutdown Bit 4 – Reserved for future BIOS Handoff Support Bits 5 – 15 – Reserved

NOTE: All reserved bits must be set to zero by the BIOS. Software developers are cautioned to ignore these bits. Attempts to use reserved bits may result in unanticipated behaviors.

4.2 ACPI WDDT Field Details

4.2.1 PCI Vendor ID

The PCI Vendor ID as assigned by the PCI-SIG (www.pcisig.org).

4.2.2 WDT Spec Version

The WDDT Specification Version 1.0 covers ICH1 through ICH5 WDT timers. The version number may be changed to reflect improvements or changes to the TCO hardware. There is no assurance that any future versions will be backward compatible.

4.2.3 WDT Descriptor Table Version

The WDDT Version is used to indicate the format of the ACPI WDDT. It may evolve as new features are added to the WDT or to the platform that need to be communicated to the OS. There is no assurance that any future revisions of the table will be backward compatible.

4.2.4 WDT Base Address

The base address of the WDT described using the Generic Address Structure as defined in Section 5.2.3.1 of the ACPI 2.0 Specification.

4.2.5 Timer Max Count

The maximum count value the WDT can accept. The WDT's maximum timeout value is the *timer maximum count* multiplied by the *timer count period*.

4.2.6 Timer Min Count

The minimum count value the WDT's count register can accept. Attempts to use any value less than this may result unanticipated behavior up to and including failure of the system to boot.

4.2.7 Timer Count Period

The effective period of the WDT count register in milliseconds. This value may not be the same as the WDT's native count, but must correspond to the value loaded into the WDT initial value register.

4.2.8 Status

This 16-bit field is used to pass information to the OS reflecting both the WDT and the platform's status.

4.2.8.1 WDT Available

This bit is used to indicate that the WDT is available. When this bit is set to zero, the software cannot use the WDT because it has been disabled. No method is supported to directly re-enable the WDT logic.

The methods the platform can use to disable the timer can range from the hardware strapping option to code in the BIOS or SMI handler that always sets the NO_REBOOT bit. This feature provides flexibility to the platform vendor and may be exposed to the user through a BIOS setup option.

4.2.8.2 WDT Active

This bit is used to indicate if the WDT is passed to the boot loader either running (active) or halted. Typically the WDT will be passed in a halted state and it is the responsibility of the OS loader or OS to take control by starting the WDT. Passing the WDT in a halted state will ensure that a user booting from an alternate media (i.e. floppy or CD) to an OS or other software that does not support the WDT, will not experience an unexpected WDT generated reset and that the software will run as expected.

There are cases when the platform and OS are tightly bound, such as in a blade server, where the platform vendor wants to use the WDT to monitor the OS load operation. This can be done by passing the WDT active, (i.e. setting the WDT *active* bit) and then depending on the OS to take control before the WDT expires.

4.2.8.3 Reserved: Ownership

This bit is reserved for future use to indicate the ownership of the WDT. This bit is valid only if the BIOS Handoff Support bit is set in the Capability field. This bit should be 0 for this revision of the specification.

4.2.8.4 User Initiated Reset Event

This bit indicates the platform was reset due to some type of user-initiated reset event. This could be due to the user pressing a reset button.

4.2.8.5 WDT Event

This bit indicates the platform was reset due to a WDT timeout event.

4.2.8.6 Power Fail Event

This bit indicates the platform suffered an unexpected power event that resulted in the platform being reset.

4.2.8.7 Unknown Reset Event

This bit indicates the platform was reset due to some type of reset event that was not initiated by the end user, by the WDT logic, or by a power failure. This bit might be set if the BIOS is unable to determine the cause of the reset.

4.2.9 Capability

This 16-bit field is used to pass information to the OS reflecting both the WDT and the platform's capability.

4.2.9.1 Auto Reset Support

This bit is set to indicate that the WDT supports reset. The reset is caused directly by the WDT logic.

4.2.9.2 Alert Support

This bit is set to indicate that the WDT supports alerting when the timeout occurs. In this case, the ICH will send alerts through its LAN interface to an external management system. The external management system can evaluate the alert message and determine if the platform should be reset or shut down.

This bit should be set only if the NO_REBOOT bit is also set.

4.2.9.3 Reserved: Immediate Shutdown Support

This bit is reserved for future use to indicate that the WDT supports immediate shutdown. When the timer times out, the platform unconditionally shuts down. The platform may not have the ability to either log or send alert messages if this feature is used.

Note: Intel's ICH product family does not currently support Immediate Shutdown capability.

4.2.9.4 Reserved: Platform Directed Shutdown Support

This bit is reserved for future use to indicate that the WDT supports Platform Directed Shutdown. In this case, after the WDT logic concludes that the platform should be reset, the platform hardware or firmware is first given the option to store parameters, send alerts, perform diagnostics, boot from an alternate partition etc. The platform may eventually reach a shutdown state.

Note: Intel's ICH product family does not currently support Platform Directed Shutdown capability.

4.2.9.5 Reserved: BIOS Handoff Support

This bit is reserved for future indication of BIOS support for a handoff mechanism. If this bit is 0, then the Ownership bit in the Status field must always be 0.

This page is intentionally left blank.

5 BIOS Requirements

At a minimum the system BIOS must incorporate the ACPI WDDT Descriptor Table and an SMI handler. This section describes both the required and recommended features. The list of recommended features is not exhaustive, but represents many common cases.

5.1 Boot Process

The BIOS has sole responsibility for the WDT during the boot process. In typical operation, the BIOS will halt the WDT before passing control to the OS or OS loader.

5.1.1 BIOS initializes TCO (required)

The BIOS must initialize the WDT within 4.8 seconds after the platform is reset. If the BIOS fails to do this in a timely manner, the WDT will cause the platform to reset. The BIOS should not enable the WDT to cause an SMI until the SMI handler has been loaded.

5.1.2 BIOS Monitors Boot Process (optional)

The WDT may be used to check for BIOS lockups. To do this, the BIOS must periodically reload the timer to keep it from reaching 0. If the WDT does reach 0, the logic will cause SMI to go active.

5.1.3 BIOS Passes the WDT to the OS Stopped (recommended)

Prior to handing control to the OS loader, it is recommended that the BIOS halt the WDT to prevent it from unintentionally resetting the system. This may be accomplished by reloading the WDT then setting the Timer Halt bit.

5.1.4 BIOS Creates WDDT (required)

Prior to handing control to the OS loader, the BIOS must ensure the ACPI WDDT Descriptor Table is correctly filled in to reflect the correct WDT capabilities and status.

5.1.5 BIOS Loads SMI Handler (required)

The BIOS must have an SMI handler that responds to a WDT timeout. The minimal SMI hander must do the following:

- Set the TCO_TMR register to its minimum allowable value
- Write to the TCO_RLD register
- Stall in SMI handler

This allows the WDT to reset the system.

If the WDT is disabled in the BIOS setup, the SMI handler may set the NO_REBOOT bit, halt the WDT and perform other actions on the WDT to prevent an undesired system reset.

5.2 BIOS Setup Features

The BIOS may have setup features that allow the user to modify the behavior of the WDT.

5.2.1 BIOS Disables WDT (required)

A BIOS that supports the use of the WDT must have a setup option that allows the user to select a mode of operation that prevents the WDT from operating even if commanded by system software. This may be accomplished by adding code to the SMI handler that is invoked only when this option is selected. If used, the SMI handler would set the NO_REBOOT bit and also halt the WDT. This effectively prevents the WDT from ever resetting the system. The BIOS must set the WDT available bit to 0 in the ACPI WDDT Descriptor Table.

5.2.2 BIOS Passes WDT to OS Running (optional)

The BIOS may allow the user the option to pass the WDT to the OS in a running state. This option may include a feature that allows the user to select the WDT timeout value. If it doesn't, the BIOS should set the WDT timeout period to its maximum value.

5.2.3 BIOS Response to WDT Reset (optional)

The BIOS may allow the user to select the action taken in the event of WDT expiration. This action may include any or all of the following: restarting the system, shutting down the system, logging the event, running diagnostics, sending alerts on LAN messages, attempting to boot from alternate partitions, or any other action that the platform vendor wishes to support.

5.3 BIOS Resume Process

When waking from an S3, S4, or S5 state, the BIOS is assumed to be given control first. The BIOS may optionally use the WDT to monitor its own progress through the resume cycle. The BIOS must pass the WDT back to the OS in the same state it reports in the WDT Active bit. For example, it should be running if the bit is set or halted if the bit is cleared.

5.4 Disabling the Watchdog Timer

It is possible to permanently disable the WDT from resetting a system if it is not going to be used.

- Software may set the HALT bit to prevent the timer from reaching zero. This prevents the WDT from generating an SMI.
- Software may set the NO_REBOOT bit. This prevents the WDT from rebooting the platform if it reaches zero. To be safe, the SMI handler should set the NO_REBOOT bit. This prevents errant software from clearing the bit and thus causing a reset.
- The hardware strap may be used to prevent reboots entirely.