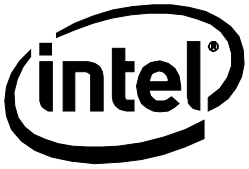


82546GB Gigabit Ethernet Controller

Specification Update

October 15, 2012

The *82546GB* Gigabit Ethernet Controller may contain design defects or errors known as errata that may cause the product to deviate from published specifications. Current characterized errata are documented in this Specification Update.

CONTENTS

REVISION HISTORY

82546GB Gigabit Ethernet Controllers Specification Update

Date of Revision	Description
April 1, 2004	Initial public release. Added errata # 4 – 6, spec change # 3, and spec clarifications # 4 – 5.
August 3, 2004	Added Errata # 7 – 9.
January 10, 2005	Added Errata #10 – 11 and spec clarification #6. Removed Spec Changes #1 and 2. Removed Spec Clarifications # 1 and 5.
April 25, 2005	Added Component Marking Information for Lead-Free Devices. Removed Specification Change #3. Removed Specification Clarifications #3, #4, and #6.
July 7, 2005	Corrected spelling of RMCP in Erratum #7. Added Erratum #12.
October 6, 2005	Added Errata #13 and #14. Added Specification Clarifications #1.
June 6, 2006	Added Erratum #15.
May 27, 2008	Added Errata #16, #17, and #18.
October 13, 2008	Updated erratum #16.
August 17, 2009	Updated section " COMPONENT IDENTIFICATION VIA PROGRAMMING INTERFACE" (changed revision to 03h).
November 11, 2009	Added Specification Clarification #2.
October 1, 2010	Updated Erratum #10.
January 16, 2012	Added Erratum #19.
October 15 2012	Added Erratum #20.

PREFACE

This document is an update to published specifications. Specification documents for these products include:

- 82546GB Gigabit Ethernet Controller Datasheet.
- 82545EM, 82545GM, 82546EB, 82546GB Gigabit Ethernet Controllers Design Guide, Ap-439.
- 8254x Family of Gigabit Ethernet Controllers Software Developer's Manual + Appendices for 82546GB, 82545EM, 82540EM, 82544EI/GC

This document is intended for hardware system manufactures and software developers of applications, operating systems or tools. It may contain Specification Changes, Errata, and Specification Clarifications.

All 82546GB product documents are subject to frequent revision, and new order numbers will apply. New documents may be added. Be sure you have the latest information before finalizing your design.

NOMENCLATURE

Specification Changes are modifications to the current published specifications. These changes will be incorporated in the next release of the specifications.

Errata are design defects or errors. Errata may cause device behavior to deviate from published specifications. Hardware and software designed to be used with any given stepping must assume that all errata documented for that stepping are present on all devices.

Specification Clarifications describe a specification in greater detail or further highlight a specification's impact to a complex design situation. These clarifications will be incorporated in the next release of the specifications.

Documentation Changes include typos, errors, or omissions from the current published specifications. These changes will be incorporated in the next release of the specifications.

COMPONENT IDENTIFICATION VIA PROGRAMMING INTERFACE

82546GB controller steppings will be identified by the following register contents:

Stepping	Vendor ID	Device ID	Revision Number
82546GB A0	8086h	1079h	03h
-	-	-	-

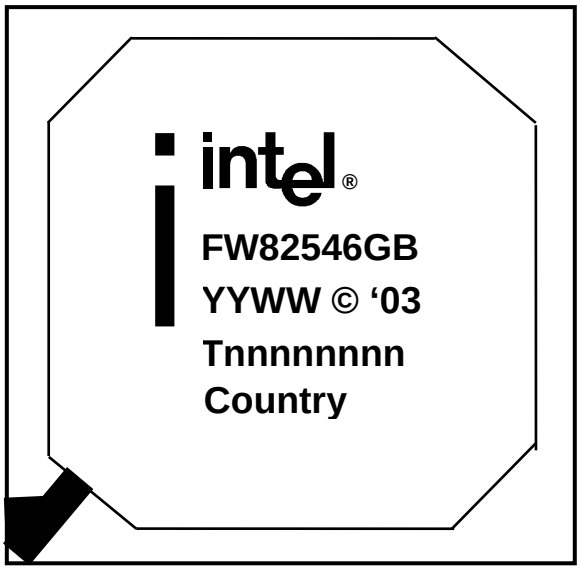
These devices also provide identification data through the Test Access Port.

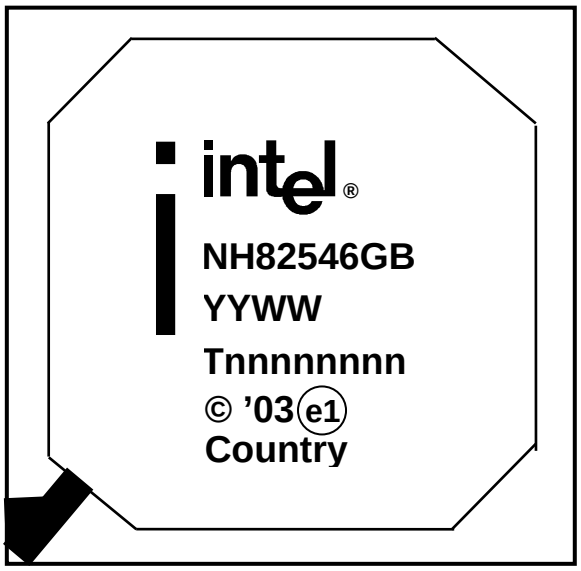
GENERAL INFORMATION

This section covers the 82546GB devices.

82546GB COMPONENT MARKING INFORMATION

Product	Stepping	QDF Number	Top Marking	Notes
82546GB	A0	Q783	FW82546GB	Engineering Samples
82546GB	A0	Q541	NH82546GB	Engineering Samples (Lead-Free)
82546GB	A0	-	FW82546GB	Production
82546GB	A0	-	NH82546GB	Production (Lead-Free)





Lead-Free Component Marking

SUMMARY TABLE OF CHANGES

The following table indicates the Specification Changes, Errata, Specification Clarifications or Documentation Changes, which apply to the listed 82546GB steppings. Intel intends to fix some of the errata in a future stepping of the component, and to account for the other outstanding issues through documentation or Specification Changes as noted. This table uses the following notations:

CODES USED IN SUMMARY TABLES

X:	Erratum, Specification Change or Clarification that applies to this stepping.
Doc:	Document change or update that will be implemented.
Fix:	This erratum is intended to be fixed in a future stepping of the component.
Fixed:	This erratum has been previously fixed.
NoFix:	There are no plans to fix this erratum.
(No mark) or (Blank Box):	This erratum is fixed in listed stepping or specification change does not apply to listed stepping.
Shaded:	This item is either new or modified from the previous version of the document.

No.	A0	Plans	ERRATA	Page	Notes
1	X	NoFix	LSO Premature Descriptor Write Back	9	-
2	X	NoFix	XOFF from Link Partner can Pause Flow-Control (XON/XOFF) Transmission	9	-
3	X	NoFix	Transmit Descriptor use of RS for non-data (Context & Null) Descriptors	9	-
4	X	NoFix	Message Signaled Interrupt Feature May Corrupt Write Transactions	10	-
5	X	NoFix	Link Establishment or Communication Problems in Fiber Mode When Link Partner Does Not Fully Comply with the IEEE 802.3 Specification	10	-
6	X	NoFix	Wakeup Packet Memory (WUPM) cleared upon reset	11	-
7	X	NoFix	Unexpected RMCP ACK packets in ASF mode	11	-
8	X	NoFix	Exceeding PCI Power Management Specification Limit of 375 mA current during reset and power state transitions	11	-
9	X	NoFix	Inbound and Outbound reads not fully decoupled in PCI-X mode	11	-
10	X	NoFix	Hang in PCI-X systems due to 2k buffer overrun during transmit operation	12	New Work Around
11	X	NoFix	CRC Errors due to Rate Adaptation FIFO Overflow in Fiber Mode	12	-
12	X	NoFix	PCI-X Arbitration Interaction with Particular Bridges Can Result in Controller Hang	13	-
13	x	NoFix	Transmit Descriptors May Be Written Back to Host, Even Without the RS Bit Set	13	-
14	x	NoFix	Legacy Transmit Descriptor Write-Back May Occur Before the Packet Data Associated with the Descriptor is Fetched	13	-
15	X	NoFix	PCI-X Burst Write Transactions to Memory Mapped Registers at Non-Qword-Aligned Offsets Fail	14	-
16	X	NoFix	Transmit (Tx) Hang on GMII/MII Interface at 10/100 Mb/s Speed	14	-
17	X	NoFix	Missing Receive (Rx) Interrupt When Using RDTR and RADV Registers	15	-
18	X	NoFix	No PCI-X Transmit (Tx) or Receive (Rx) DMA Activity From the 82546GB	15	-
19	X	NoFix	SerDes: RXCW.RxConfigInvalid Set Incorrectly	16	-
20	X	NoFix	Short Packet Data Corruption	16	-

No.	A0	Plans	SPECIFICATION CLARIFICATIONS	Page	Notes
1	X	NoFix	82546GB Ports Can be Disabled Individually	16	-
2	X	NoFix	82546GB Software Device Drivers in Release 14.7 Are the Final WHQL Drivers	16	-

SPECIFICATION CHANGES

No specification changes reported at this time**.**

ERRATA

1. LSO Premature Descriptor Write Back

Problem: For large send fetches ONLY (not normal or jumbo frames) the internal DMA engine will decompose the large-send data fetch into a series of individual requests that are completed sequentially. When all read data associated with the first internal DMA request has been fetched, the descriptor is flagged as ready for writeback. Though all data associated with the entire LSO descriptor will eventually be fetched, the descriptor writeback may occur prematurely. The device should wait until all bytes associated with the data descriptor have been completely fetched before writing back the transmit descriptor.

Implication: Due to premature write back, an operating system may release and reallocate the buffer, potentially causing buffer re-use and transmission of incorrect data.

Workaround: Utilize a second descriptor to point to the last four bytes of the large-send transmit data, and ensure that the buffer is not freed to the operating system/application until the second descriptor has been marked as complete via a status writeback operation.

Status: Intel does not plan to resolve this erratum in a future stepping of the 82546GB Gigabit Ethernet Controller.

2. XOFF from Link Partner can Pause Flow-Control (XON/XOFF) Transmission

Problem: When the 82546GB transmitter is paused (by having received an XOFF from link partner), not only is the transmit of normal packets paused, but also of outbound XON/XOFF frames resulting from Receive Packet Buffer levels and Flow-Control Thresholds. Normally, partner's XOFF packets only pause the LAN controller for a finite time interval, after which outbound XON/XOFF's due to Receive Packet-Buffer fullness are again permitted to be sent.

Implication: If the transmitter is paused when a Receive FIFO XOFF threshold is reached, the transmission of XOFF frames does not occur and Receive FIFO overrun may potentially occur, resulting in lost packets. This is only expected to be seen with an abnormally high pause time from link partner's XOFF packet(s).

Workaround: Receive Flow-Control Thresholds may be tuned/lowered based on the expected maximum pause interval expected from link partner's XOFF packet in order to minimize the likelihood of Receive FIFO overruns.

Status: Intel does not plan to resolve this erratum in a future stepping of the 82546GB Gigabit Ethernet Controller.

3. Transmit Descriptor use of RS for non-data (Context & Null) Descriptors

Problem: Due to an internal logic error in the descriptor internal queue, if the internal descriptor queue becomes completely full of pending descriptor status writebacks, the descriptor logic may issue a writeback request with an incorrect writeback amount. The internal descriptor queue may accumulate pending writebacks if transmit descriptors that do not directly refer to transmit data buffers (e.g. context or Null descriptors) are submitted with a status-writeback request (RS asserted) and legacy writeback (status byte writeback only) is utilized.

Implication: Due to the invalid internal writeback request size, the PCI logic may hang.

Workaround: Ensure that status-writeback reporting (RS) is not set on context or Null descriptors. Alternatively, utilize full-descriptor writebacks (TXDCTL.WTHRESH >= 1). The former workaround is the recommended alternative.

Status: Intel does not plan to resolve this erratum in a future stepping of the 82546GB Gigabit Ethernet Controller.

4. Message Signaled Interrupt Feature May Corrupt Write Transactions

Problem: The problem is with the implementation of the Message Signaled Interrupt (MSI) feature in the Ethernet controller. During MSI writes, the controller should use the MSI message data value in PCI configuration space.

At the same time, for normal write transactions (received packet data and/or descriptor writebacks), the controller temporarily stores the data for write transactions in a small memory until it is granted ownership of the PCI/PCI-X bus. The error condition occurs when during the MSI operation the controller incorrectly pulls data from the memory storing the data waiting to be written. If there are any write transactions waiting when this occurs, these transactions may become corrupted. This, in turn, may cause the network controller to lock up and become unresponsive.

Implication: If the affected products are used with an OS that utilizes Message Signal Interrupts and no accommodations are made to mitigate the use of these interrupts, data integrity issues may occur.

Workaround: For PCI systems, advertisement of the MSI capability can be turned off by setting the MSI Disable bit in the EEPROM (Init Control Word 2, bit 7).

For PCI-X systems where MSI support is enumerated as part of the PCI-X specification, Intel is working with OS vendors to ensure that any future implementations of their operating systems can detect these products and avoid using the MSI mechanism. Further details will be communicated as they become available.

Status: Intel does not plan to resolve this erratum in a future stepping of the 82546GB Gigabit Ethernet Controller.

5. Link Establishment or Communication Problems in Fiber Mode When Link Partner Does Not Fully Comply with the IEEE 802.3 Specification

Problem: The following minor compliance issues have been discovered between the TBI/SERDES mode symbol synchronization logic and the IEEE specification:

- When presented with short sequences of malformed code groups, the receive synchronization logic within the Ethernet controller may acquire & indicate link/synchronization prematurely or incorrectly

- When presented with certain short sequences of malformed code groups, the logic may retain link/synchronization indication through the error sequence instead of immediately detecting and dropping link/synchronization

- With some specific erroneous sequences of code groups, the auto-negotiation logic may establish link in certain very specific situations where the specification says it should not

- Finally, the receive error detection logic may not detect and count some symbol errors when malformed idle patterns are received.

Implication: If a link partner is not compliant with the IEEE 802.3 Specification in certain very specific ways, the 82546GB controller may not be able to establish link or communicate properly with it. If the controller is tested for strict compliance with the IEEE 802.3 Specification, it may fail some of the Clause 36 and Clause 37 test cases.

However, Intel has performed extensive compatibility testing as an integral part of controller HW validation, and continues to do so with the latest Ethernet devices. To date, these issues have not been shown to cause interoperability problems with any Ethernet devices currently in production.

Workaround: None.

Status: Intel does not plan to resolve this erratum in a future stepping of the 82546GB Gigabit Ethernet Controller.

6. Wakeup Packet Memory (WUPM) cleared upon reset

Problem: The 82546GB specifications state that the Wakeup Packet Memory (WUPM) is not cleared on any reset. This is incorrect. Any reset or power-state transition will clear the contents of these registers.

Implication: Because a power-state transition takes place on wakeup, the Wakeup Packet Memory will always be cleared before it can be read by software. This makes the memory effectively unable to provide the capability for inspecting the wakeup packet content.

Workaround: There is no workaround. WUPM will be considered to be defeatured for the affected controllers.

Status: Intel does not plan to resolve this erratum in a future stepping of the 82546GB Gigabit Ethernet Controller.

7. Unexpected RMCP ACK packets in ASF mode

Problem: According to the RMCP protocol, the response to all RMCP commands (except ACK) should be an RMCP ACK packet. In ASF mode, the Ethernet Controller responds to RMCP ACK packets with a second ACK.

Implication: Any management software should be aware of this behavior and not respond to the additional RMCP ACK packets.

Workaround: None.

Status: Intel does not plan to resolve this erratum in a future stepping of the 82546GB Gigabit Ethernet Controller.

8. Exceeding PCI Power Management Specification Limit of 375mA current during reset and power state transitions

Problem: During resets and power state transitions the controller may briefly draw more than 375 mA of current as the digital signal processors in the PHY attempt to converge. The excessive current draw persists for approximately 100 milliseconds. Refer to the "Power Specifications -- MAC/PHY" section of this document for specific values.

Implication: If an application has current limiting circuitry in place, the Ethernet Controller may trigger these safeguards in power-up or during transitions between D0 and D3 power states.

Workaround: None.

Status: Intel does not plan to resolve this erratum in a future stepping of the 82546GB Gigabit Ethernet Controller.

9. Inbound and Outbound reads not fully decoupled in PCI-X mode

Problem: If the Ethernet controller receives a read as a target and signals a split response it will not deliver a completion to this read until its entire outstanding read requests have been satisfied. The device should not make the completion of a sequence for which it is the completer contingent upon another device completing a sequence for which it is a requester.

Implication: There is a slight system performance impact due to this erratum. Processors may be stalled while the read transaction is outstanding, so the extra delay may adversely affect CPU utilization.

If and only if a host bridge also has a similar dependency, the possibility of a deadlock exists. A situation may arise where the bridge is waiting for the controller to respond to a DWord read while the controller is waiting for the bridge to complete a block read.

Workaround: None.

Status: Intel does not plan to resolve this erratum in a future stepping of the 82546GB Gigabit Ethernet Controller.

10. Hang in PCI-X systems due to 2k Buffer Overrun during Transmit Operation

Problem: This Ethernet device has an error in the way that it stores data from PCI-X read transactions. If the controller is operating in PCI-X mode and its read data FIFO fills completely then the device can miscalculate the amount of free space in the FIFO and lose all of this data.

This erratum does not apply to devices running in PCI mode only.

Implication: If this device enters this erratum state, the chip loses 2 kilobytes of data. The transmit and receive units of the chip will hang waiting for this data which will never arrive. No data will be corrupted. Once this has occurred, a reset is required to restore the device to normal operation.

If using larger MTUs (jumbo frames), the chance of reaching this erratum state also increases.

Workaround: The issue can occur only when one packet is being completed and the next being started. Therefore, if the first fragment of every packet is limited in size the overflow can be prevented entirely. Drivers can work around this issue by ensuring that the size in the first descriptor of every packet less than 2016 bytes.

Additionally, if the Maximum Memory Read Byte Count field in the PCI-X Command Register is set to 00b (512 bytes), the number of descriptors in use (tail minus head) in any individual queue should not exceed 32.

Status: Intel does not plan to resolve this erratum in a future stepping of the 82546GB Gigabit Ethernet Controller.

11. CRC Errors due to Rate Adaptation FIFO Overflow in Fiber Mode

Problem: In TBI mode and internal-SERDES mode this Ethernet device uses a small FIFO in its receive path to compensate for minute differences between the speed of the link partner's clock and the device's local clock. If the link partner has a faster clock, this FIFO will fill slowly during a packet, and then drain during inter-frame gaps.

The device has an error in the way that this FIFO empties, causing it to wait several cycles into the inter-frame gap before it begins recovering clock drift.

This only occurs during operation in fiber mode. Internal PHY mode used for copper applications is unaffected by this erratum.

Implication: If the Ethernet device is linked to a partner with a substantially faster clock and multiple frames arrive in sequence with minimal inter-frame spacing, then the device may not have time to recover all of the accumulated drift between frames. The synchronization FIFO will overflow and drop 4 bytes of the packet, which will be visible as a CRC error.

The larger the difference between the link partner's clock and the Ethernet controller's clock, the fewer back-to-back frames need to be received to see CRC errors. In practice, this will be a very rare occurrence for two reasons. First, most Ethernet devices use clock frequencies near the center of the allowed range, so the difference between clocks will be small. Second, long strings of packets with minimal inter-frame spacing are rare on most networks.

Workaround: This erratum may be worked around by setting a larger inter-frame spacing. Specifically, switches must be configured to an inter-frame gap of at least 144 ns (18 symbols) for MTUs less than 10,000 bytes or at least 160 ns (20 symbols) for MTUs between 10,001 and 16,000 bytes.

Alternatively, a new board design could use a reference clock source with a frequency near the high end of the 802.3 standard's allowed range. This would create a situation where the only way to trigger the erratum was for the link partner to have a faster clock which would violate the 802.3 standard.

Status: Intel does not plan to resolve this erratum in a future stepping of the 82546GB Gigabit Ethernet Controller.

12. PCI-X Arbitration Interaction with Particular Bridges Can Result in Controller Hang

Problem: In PCI-X mode, the 82546GB arbitration logic de-asserts its REQ# signal briefly before beginning a transaction. This is permitted by the PCI-X specification but is not optimal behavior.

Implication: Under a very specific set of circumstances, certain PCI-X bridge components might respond to this behavior in a way that hangs the system. If the PCI-X bridge sees the 82546GB de-asserting REQ# then it might choose to immediately de-assert GNT#. The 82546GB will then abandon its attempt to begin a transaction and re-start its arbitration cycle. If this cycle happens repeatedly, the 82546GB might be unable to initiate PCI-X transactions.

The 82546GB interoperates well with the majority of PCI-X bridge components. However, the Intel® 31154 PCI-X-to-PCI-X bridge exhibits the behavior described above. Systems that use both components in PCI-X mode will need to implement one of the following workarounds.

Workaround: This issue can be addressed in several ways:

1. Operate the bus segment containing the 82546GB in PCI mode. The erratum does not occur in PCI mode.

2. Configure the arbiter in the PCI-X Bridge to change its arbitration behavior. Changing something as small as the master on which it parks can alter timings enough to avoid the problem.

3. Add a Programmable Logic Device (PLD) to your application to detect this situation and break the deadlock by changing the duration of the 82546GB's REQ# assertion.

For further details on these options, including sample code for a Programmable Logic Device, see TA-176 ("Possible System Hang When Using an Intel® 82546GB LAN Controller with Certain PCI Bridges").

Status: Intel does not plan to resolve this erratum in a future stepping of the 82546GB Gigabit Ethernet Controller.

13. Transmit Descriptors May Be Written Back to Host, Even Without the RS Bit Set

Problem: If the RS bit is set on at least some transmit descriptors submitted to the device, it is possible that some other transmit descriptors without the RS bit set will be incorrectly written back to host memory.

Implication: The unnecessary descriptor write-backs will not cause a functional issue, but they may result in a small amount of unnecessary host bus bandwidth to be consumed.

Workaround: None.

Status: Intel does not plan to resolve this erratum in a future stepping of the 82546GB Gigabit Ethernet Controller.

14. Legacy Transmit Descriptor Write-Back May Occur Before the Packet Data Associated with the Descriptor is Fetched

Problem: If a legacy transmit operation directly follows a TCP Segmentation Offload transmit operation, the logic may incorrectly associate the successful completion of the TSO transmit with the next descriptor. If the next descriptor is a legacy descriptor, under certain timing scenarios it is possible for the legacy descriptor to be incorrectly written back to host memory with the DD bit set. This might occur even though the packet data for the legacy descriptor has not yet been fetched.

Implication: Due to the premature write back, an operating system may release and reallocate the transmit buffer, potentially causing buffer re-use or transmission of incorrect data.

Workaround: Utilize at least two descriptors for any legacy transmit operation. Do not reallocate any buffers associated with the transmit operation until the last descriptor has been written back.

Status: Intel does not plan to resolve this erratum in a future stepping of the 82546GB Gigabit Ethernet Controller.

15. PCI-X Burst Write Transactions to Memory Mapped Registers at Non-Qword-Aligned Offsets Fail

Problem: The device does not properly handle burst (greater than 4 bytes in length) write transactions to its memory mapped register space. Registers with addresses ending in 0x0 or 0x8 work, but registers with addresses ending in 0x4 or 0xC cannot be written. For example, a PCI-X Memory Write Block transaction writing 16 bytes to offset 0x2800 properly writes to locations 0x2800 and 0x2808; however, locations 0x2804 and 0x2808 are not updated.

Implication: The specification for the device states that memory-mapped registers should only be written 32 bits at a time. Software should always be written to follow this rule. It is particularly important during initialization when most of the memory-mapped register accesses take place. Unfortunately, some platforms might perform write combining, which turn consecutive 32-bit writes into a single burst transaction. In this case, the registers with addresses ending in 0x4 or 0xC are not written. Common areas for consecutive adjacent register writes include setup of the large register arrays (MTA, VFTA, and RAR).

Workaround: If there is platform-level control for a write-combining feature, turn it off.

Alternately, software can be written with the possibility of write combining in mind:

- Writes to consecutive registers can be followed by a read transaction, which should flush the posted write from any bridge that might perform combining.
- Initialization of large register arrays (VFTA, MTA) can be performed in reverse, writing the highest location first and working backward to the lowest. This prevents write combining from occurring since the writes no longer meet the rules that allow combining.

Status: Intel does not plan to resolve this erratum in a future stepping of the 82546GB Gigabit Ethernet Controller.

16. Transmit (Tx) Hang on GMII/MII Interface at 10/100 Mb/s Speed

Problem: Internally, a Tx hang occurs as a result of a synchronization failure when signals pass through two clock domains. This failure leads to a secondary problem where an incorrect byte count gets latched for an upcoming transmit operation. The incorrect byte count hangs the Tx logic. Note that any Receive (Rx) activity is unaffected.

This failure is only seen when using the 82546GB Gigabit Ethernet Controller at 10/100 Mb/s speeds. It occurs whether the device is configured with an internal or external PHY. The failure only occurs in MII mode (10/100 Mb/s) because the GMII mode (1000 Mb/s) bus uses a different clocking scheme.

Implication: When the failure occurs, all transmit activity out of the 82546GB halts.

Workaround: The workaround must be performed at the driver level. For customers developing their own driver, a workaround for avoiding this failure is to submit single Tx packets to the 82546GB, one at a time. Meaning, submit a packet and wait until that packet has been completely transmitted out on the wire before submitting a new packet. This is accomplished by verifying that the head and tail pointers for the Tx descriptor FIFO are equal to each other, and the head and tail pointers for the Tx FIFO are also equal to each other before submitting a new packet. The algorithm is:

```
If (  (Tx descriptor head = Tx descriptor tail)  AND
          (Tx FIFO head (saved) =  Tx FIFO tail)  )
                Then submit next Tx packet
```

Register addresses are:

- Tx descriptor head (TDH) - 0x3810
- Tx descriptor tail (TDT) - 0x3818
- Tx FIFO head saved (TDFHS) - 0x3420
- Tx FIFO tail (TDFT) - 0x3418

Status: Intel does not plan to resolve this erratum in a future stepping of the 82546GB Gigabit Ethernet Controller.

17. Missing Receive (Rx) Interrupt When Using RDTR and RADV Registers

Problem: This issue results in a loss of an Rx interrupt and/or a large delay between the time a packet is sent to the 82546GB and later received in host memory.

Primarily, there is a large delay between the time an Rx packet is sent to the 82546GB and the time it is received in host memory. This delay occurs randomly and infrequently; however, no packet loss occurs.

Internally, the delay occurs because the Receive Delay Timer (RDTR) or the Absolute Delay Timer (RADV) is cleared at an incorrect time relative to a receive descriptor write-back operation. Both timers were designed to be cleared automatically during write-backs to help prevent spurious interrupts. This specific issue occurs because both timers are cleared after a write-back, instead of before. As a result, if a new packet arrives during a write-back, the timers get cleared after the new packet arrives. Therefore, there is no descriptor write-back and no interrupt generated to indicate the arrival of the new packet until another (newer) packet is received and re-starts the timers. Then, after the timers expire, the descriptors for both the stuck packet and newer packet are written back to host memory, and the interrupt bit is set. In summary, packets are not lost, just delayed.

Implication: No packet loss occurs. However, there might be a large delay between the time an Rx packet is received in the 82546GB and the time the descriptor is written back to memory, and finally an interrupt generated.

Workaround: It is recommended that the RDTR and RADV registers not be used for moderating Rx interrupts. The preferred solution is to use the Interrupt Throttling Register (ITR). Refer to the PCI/PCI-X Family of Gigabit Ethernet Controllers Software Developer's Manual for ITR details.

Status: Intel does not plan to resolve this erratum in a future stepping of the 82546GB Gigabit Ethernet Controller.

18. No PCI-X Transmit (Tx) or Receive (Rx) DMA Activity From the 82546GB

Problem: This issue typically results in a system hang, Tx hang, stall, or a PCI-X bus stuck condition. The most notable observation is that all transmit (Tx) and receive (Rx) DMA activity stops. However, the PCI-X bus itself is not actually hung and reads/writes to/from the 82546GB registers are still functional. Programmers might also notice an accumulation of unprocessed Tx descriptors in host memory. Meaning, Tx descriptors have been submitted to the 82546GB, but the corresponding descriptor *Done (DD)* bit has not been set.

Internally, this issue occurs as a result of a logic fault in the DMA engine. As transmit and receive requests are processed, the read/ write pointers for the engine's Job FIFO reach a state where they equal each other and the failing logic interprets this as a FIFO empty condition. The reality is that the FIFO is actually full. As a result, the FIFO is deadlocked and no transmit or receive jobs can execute. All DMA operations stop.

The FIFO empty condition is extremely timing specific. It is affected by the types of transactions (R/W), size of memory transactions, sequence of transactions, host memory latencies, and host bus speeds. The FIFO empty condition might also be seen for one configuration (such as, payload size = 1500 bytes) and not another (such as, payload size = 495 bytes). Again, the most notable observation is that no Tx and Rx activity is seen on the PCI-X bus.

Implication: This is a major failure in the DMA engine logic that completely stops all DMA activity into or out of the 82546GB; however, PIO operations (register R/W) are unaffected.

Workaround: The workaround must be performed at the driver level. Specifically, a driver needs to include a hang detect, MAC reset function as follows:

1) Detect that the DMA engine is in fact hung.
2) Reset the MAC portion of the 82546GB to get out of the hung state.

The workaround needs to include the following actions:

Hang detect:

- Create a code loop to detect uncompleted transmits after a fixed amount of time (per transmit). There are two methods that can be used for detection:
    1) Periodically monitor the *DD* bit in the transmit descriptors and watch for incomplete descriptors.
    2. Periodically monitor the Transmit Descriptor Head (TDH) register and watch for a head not advancing condition (last value = current value).

Note: A reasonable value for periodically monitoring either condition is two seconds (at 1 Gb/s speed). Programmers might have to adjust this value for their specific system configuration.

After a hang:

- Disable interrupts (0xFFFFFFFF -> IMC).
- Disable receives (0x0 -> RCTL).
- Disable transmits (0x0 ->TCTL).
- Delay for [at least] 100 milliseconds.
- Finish receiving any data in host memory back to the operating system.

MAC reset:

- Reset the MAC (set CTRL.RST).
- Delay for 200 ms.
- Clear the transmit ring.
- Clear the receive ring.
- Initialize the Rx descriptor ring length; RDLEN.
- Initialize the Rx descriptor head pointer; RDH.
- Initialize the Rx descriptor tail pointer; RDT.
- Initialize the Tx descriptor ring length; TDLEN.
- Initialize the Tx descriptor head pointer; TDH.
- Initialize the Tx descriptor tail pointer; TDT.
- Initialize the Rx control register; RXDCTL.
- Initialize the Tx control register; TXDCTL.
- Initialize the Interrupt Throttling Register; ITR.
- Initialize the Transmit Control Register; TCTL (do not set the *EN* (enable) bit yet).
- Initialize the Receive Control Register; RCTL (do not set the *EN* (enable) bit yet).
- Read the PHY speed/duplex settings.
- Write the CTRL register with the PHY's speed/duplex settings.
- Wait for link up.
- Initialize interrupt settings in Interrupt Mask Register; IMS.
- Read and clear the Interrupt Cause Register; ICR.
- Enable the receiver (RCTL.EN = 1b).
- Enable the transmitter (TCTL.EN = 1b).

The 82546GB is now ready to receive and transmit packets.

Status: Intel's drivers already incorporate the hang detect, device reset function, which is more than what is actually required to workaround this issue. Custom drivers must include the previously stated workaround for this specific erratum.

19. SerDes: RXCW.RxConfigInvalid Set Incorrectly

Problem: When the device has been receiving a continuous stream of /C/ ordered sets for an extended period of time, the RXCW.RxConfigInvalid might be set as the result of an internal FIFO overflow even if all the input symbols are valid.

Implication: False indication of invalid symbols may cause the driver to disable the link when there is really no problem.

Workaround: Software that uses the RxConfigInvalid bit should account for this behavior. For example, when the RxConfig bit is consistently 1b, it would be reasonable to ignore the RxConfigInvalid bit.

Intel drivers address this erratum for the device by looking to see if the 82546GB has Sync and Invalid bit is set then read RXCW several times, if Sync and Config both are consistently 1 then ignore Invalid bit and restart Autoneg. This is done when link is down and driver is trying to determine if the link support Auto Negotiation by looking for /C/ ordered set and if /C/ ordered sets are seen then Auto Negotiation is enabled (TXCW.ANE) to try an link up via Auto Negotiation.

Status: Intel does not plan to resolve this erratum in a future stepping of the 82546GB Gigabit Ethernet Controller.

20. Short Packet Data Corruption

Problem: Due to the 82546GB's padding logic for short packets (< 60 bytes), the contents of these packets might become corrupted. Typically, an additional 8 bytes can be seen in a corrupted packet.

Implication: Corrupted data in short packets.

Workaround: Software device drivers should pad all short packets out to 60 bytes.

Status: Intel drivers implement the recommended workaround.

SPECIFICATION CLARIFICATIONS

1. 82546GB Ports Can be Disabled Individually

Clarification: The 82546GB allows ports to be disabled individually. However, if only one port is disabled (using the LAN disable feature) while the other port is left operational, care must be taken to ensure that the SMBus feature is NOT enabled on the disabled port. If the SMBus is left enabled on a disabled port the 82546GB may prevent the system from booting.

2. 82546GB Software Device Drivers in Release 14.7 Are the Final WHQL Drivers

Clarification: Any changes to the current 82546GB Windows software device driver will no longer be WHQLd. Windows 7 and Windows 2008 Server R2 will only provide in-box base driver support. There will be no support for any future Windows operating systems.

Note that Linux kernel 2.6.31 is the last kernel that is validated for the 82546GB.

DOCUMENTATION CHANGES

No documentation changes reported at this time**.**