Working Draft

T13 1510D

Revision 0a
7 August 2001

ATA Host Adapter Standards

T13 Technical Editor:

Tony Goodfellow
Pacific Digital Corporation
2052 Alton Parkway
Irvine, CA92602
USA

Tel: 949-252-1111
Fax: 949-252-9397
Email: tgoodfellow@pacificdigital.com

Some technologies and techniques, which are covered in this document, are the subject of one or more patent applications.

Other Points of Contact:

T13 -Chair
Pete McLean
Maxtor Corporation
2190 Miller Drive
Longmont, CO 80501
Tel: 303-678-2149
Fax: 303-682-4811

T13 –Vice Chair
Dan Colegrove
IBM Corporation
Tel: 702-614-6119
Fax: 702-614-7955

NCITS Secretariat
Administrator Standards Processing
1250 Eye Street, NW Suite 200
Washington, DC 20005
Tel: 202-737-8888
Fax: 202-638-4922
Email: NCITS@ITIC.NW.DC.US

T13 Reflector
???????

T13 Web Site
????http://www.t13.org

T13 Anonymous FTP Site
??????

T13 mailings
Global Engineering
15 Inverness Way East
Englewood, CO 80112-5704
Tel: 303-792-2181 or 800-854-7179
Fax: 303-792-2192

DOCUMENT STATUS

Revision 0a - 7 August 2001

ANSI®
NCITS.*-xxxx**

American National Standard
for Information Systems —

ATA Host Adapter Standards

Secretariat
Information Technology Industry Council

Approved mm dd yy

American National Standards Institute, Inc.

Abstract

This standard specifies the Host System Interface used to control AT Attachment Interface devices It provides a common Programming interface for systems manufacturers, system integrators, software suppliers, and suppliers of intelligent storage devices.

American National Standard

Approval of an American National Standard requires verification by ANSI that the requirements for due process, consensus, and other criteria for approval have been met by the standards developer. Consensus is established when, in the judgment of the ANSI Board of Standards Review, substantial agreement has been reached by directly and materially affected interests. Substantial agreement means much more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that effort be made towards their resolution.

The use of American National Standards is completely voluntary; their existence does not in any respect preclude anyone, whether he has approved the standards or not, from manufacturing, marketing, purchasing, or using products, processes, or procedures not conforming to the standards.

The American National Standards Institute does not develop standards and will in no circumstances give interpretation on any American National Standard. Moreover, no person shall have the right or authority to issue an interpretation of an American National Standard in the name of the American National Standards Institute. Requests for interpretations should be addressed to the secretariat or sponsor whose name appears on the title page of this standard.

CAUTION NOTICE: This American National Standard may be revised or withdrawn at any time. The procedures of the American National Standards Institute require that action be taken periodically to reaffirm, revise, or withdraw this standard. Purchasers of American National Standards may receive current information on all standards by calling or writing the American National Standards Institute.

CAUTION: The developers of this standard have requested that holders of patents that may be required for the implementation of the standard, disclose such patents to the publisher. However, neither the developers nor the publisher have undertaken a patent search in order to identify which, if any, patents may apply to this standard.

As of the date of publication of this standard and following calls for the identification of patents that may be required for the implementation of the standard, notice of one or more such claims has been received.

By publication of this standard, no position is taken with respect to the validity of this claim or of any rights in connection therewith. The patent holders have, however, filed a statement of willingness to grant a license under these rights on reasonable and nondiscriminatory terms and conditions to applicants desiring to obtain such a license. Details may be obtained from the publisher.

No further patent search is conducted by the developer or the publisher in respect to any standard it processes. No representation is made or implied that licenses are not required to avoid infringement in the use of this standard.

Published by
American National Standards Institute
11 West 42nd Street, New York, New York 10036

Contents

Tables

Figures

1 Foreword

(This foreward is not part of American National Standard ***-****.)

This standard was developed by the ATA ad hoc working group of Accredited Standards Committee NCITS starting in 2001. This document includes annexes that are informative and are not considered part of the standard.

Requests for interpretation, suggestions for improvement and addenda, or defect reports are welcome. They should be sent to the NCITS Secretariat, Information Technology Industry Council, 1250 Eye Street, NW, Suite 200, Washington, DC 20005-3922.

This standard was processed and approved for submittal to ANSI by Accredited Standards Committee on Information Processing Systems, NCITS. Committee approval of the standard does not necessarily imply that all committee members voted for approval. At the time it approved this standard, the NCITS Committee had the Karen Higginbottom, Chair

(Vacant), Vice-Chair

Monica Vago, Secretary

Organization Represented Name of Representative

AMP, Inc John Hill, Charles Brill (Alt.)

Apple Computer David Michael, Jerry Kellenbenz (Alt.)

AT&T Thomas Frost, Paul Bartoli (Alt.)

Bull HN Information Systems, Inc. Patrick L. Harris

Compaq Computer Corporation Steven Heil, Seve Park (Alt.)

Eastman Kodak Michael Nier

Hewlett-Packard Karen Higginbottom, Donald Loughry (Alt.)

Hitachi America, Ltd. John Neumann, Kei Yamashita (Alt.)

Hughes Aircraft Company Harold L. Zebrack

IBM Corporation Ron Silletti, Joel Urman (Alt.)

Institute for Certification of Computer Professionals Kenneth M. Zemrowski, Tom Kurihara (Alt.)

Lucent Technologies, Inc. Herbert Bertine, Tom Rutt (Alt.)

National Communications Systems Dennis Bodson, Frack McClelland (Alt.)

National Institute of Standards and Technology Michael Hogan, Bruce K. Rosen (Alt.)

Panasonic Technologies, Inc.. Judson Hofmann, Terry J. Nelson (Alt.)

Share, Inc. David Thewlis, Gary Ainsworth (Alt.)

Sony Electronics, Inc. Masataka Ogawa, Michael Deese (Alt.)

Storage Technology Corporation Joseph S. Zajaczkowski

Sun Microsystems, Inc. Gary Robinson

Sybase, Inc. Donald Deutsch, Andrew Eisenberg (Alt.)

Texas Instruments, Inc. Clyde Camp, Fritz Whittington (Alt.)

Unisys Corporation Arnold F. Winkler, Stephen P. Oksala (Alt.)

U.S. Department of Defense/DISA Jerry L. Smith, C. J. Pasquariello (Alt.)

U.S. Department of Energy Carol Blackston, Bruce R. White (Alt.)

Xerox Corporation John B. Flannery, Jean Baroness (Alt.)

Subcommittee T13 on ATA Interfaces, that reviewed this standard, had the following members:

Pete McLean, Chairman

Dan Colegrove, Vice Chairman

ATA/ATAPI ad hoc Working Group, that developed this standard, had the following additional participants:

2 Normative References

In addition to the references below, see also Section 3: Definitions, Abbreviations, and Conventions.

2.1 Content Imported from Normative Specifications

Where material within this specification has been imported from another, normative specification, in whole or in part, it is done so for the sake of readability of this document and the normative source document is identified. The source specification has precedence where there is a difference of definition. References are listed in this section. Each reference has an abbreviated reference form, enclosed within braces '{}'. This abbreviation, braces included, is used in the body of this document. Where the reference cites a standard and working drafts, the cross-reference is to the standard that was current at the time when this document was written.

2.2 Industry Standard References

2.2.1 ANSI Information Technology – AT Attachment with Packet Interface (ATA/ATAPI-5)

Abbreviation: {ATA Spec}

Published standard: Information Technology - AT Attachment with Packet Interface - 5 (ATA/ATAPI-5), ANSI National Standard for Information Systems, # ANSI NCITS 340-2000.

Draft standard: Information Technology - AT Attachment Interface with Packet Interface Extensions - 6 (ATA/ATAPI-6), Proposed Draft # T13-1411D Revision 1b, 14 March 2001.

2.2.2 PCI Local Bus Specification

Abbreviation: {PCI Spec}

Standard for the PCI Local Bus, Revision 2.2, published December 18, 1998, by the PCI Special Interest Group (PCI SIG).

2.2.3 PCI Hot-Plug Specification

Abbreviation: {PCI Hot Plug}

Standard for the PCI Local Bus, Revision 1.0, published October 6, 1997, by the PCI Special Interest Group (PCI SIG).

2.2.4 PCI Bus Power Management Specification

Abbreviation: {PCI PMS}

Standard for the PCI Local Bus – Revision 1.1, published December 18, 1998, by the PCI Special Interest Group (PCI SIG).

3 Definitions, Abbreviations, and Conventions

3.1 Definitions and Abbreviations

3.1.1 ADMA

Automatic Direct Memory Access

3.1.2 ADMA Command Chaining

The principle objective of implementing command chaining in the ADMA hardware is to allow the device driver and the ADMA hardware to be *loosely* coupled. To do this, the software can build up a list of tasks (a command chain) for the hardware to execute. The hardware *independently* reads these requests from memory and executes the tasks. When the hardware completes a task, it interrupts the host to inform the host that the task is complete, but immediately proceeds to the next task *without waiting*.

3.1.3 ADMA Mode

An operating mode of the ADMA engine which uses ADMA command chaining.

3.1.4 ATA Adapter

An ATA adapter is the hardware that is the interface between the ATA host and the ATA Channel. The embodiment of this includes Integrated Circuits, and plug-in adapters.

3.1.5 ATA Bus

The physical connection between an ATA adapter and an ATA device, that consists of conductors carrying signals.

3.1.6 ATA Bus Release

{ATA Spec} The act of clearing both DRQ and BSY to zero, and setting ATA REL to one, before the action requested by a command is completed. This allows the host to select the other device on the channel. (Applies only to ATA devices that implement Overlap Protocol, by releasing the bus.)

3.1.7 ATA Channel

The ATA Channel is the logical transport mechanism between the ATA host and the ATA devices on an ATA Bus. Each ATA Channel may have up to two ATA devices connected to it.

3.1.8 ATA Command Acceptance

{ATA Spec} A command is written to the currently selected ATA device when the device's Status Register Busy Bit is equal to zero and the host writes to the device Command Register. The command is considered *accepted* after the command has been written, and the device's Busy Bit transitions from zero to one. An exception exists for following commands: Execute Device Diagnostic and Device Reset.

3.1.9 ATA Command Queue (in the device)

{ATA Spec} The ATA Command Queue in the device is the set of all commands that an ATA device has accepted and is currently processing.

3.1.10 ATA device

{ATA Spec} An ATA device is a data storage device. Traditionally, a device on the ATA interface has been a hard drive, but any form of storage device may be placed on the interface, provided that the device adheres to the ATA standard.

3.1.11 ATA DMA

The transfer of data between an ATA device and an ATA adapter under the control of the DMARQ and DMACK signals on the ATA bus. There are two methods of DMA defined: Multiword DMA, where the adapter controls timings, and Ultra-DMA, where the sender controls timings.

3.1.12 ATA Host

The ATA host is the host system in which the software that controls the functions of the ATA Subsystem is executed.

3.1.13 ATA Multiword DMA

ATA Multiword DMA is defined to transfer data at up to 16 MB/s. This protocol has traditionally been used in conjunction with the PCI DMA protocol to provide a more efficient means of transferring data through the system.

3.1.14 ATA-n

A shorthand reference to the standard specified in the ATA-n (or ATA/ATAPI-n, as applicable) standards document, whether published as final, circulated in draft form, or only in the planning stage.

3.1.15 ATA Subsystem

The ATA subsystem includes the ATA hardware elements, which consist of an ATA adapter, an ATA channel, and ATA device(s).

3.1.16 ATA Overlap Protocol

{ATA Spec} The ATA overlap protocol allows an ATA device to perform an ATA Bus Release, so that commands may be executed by another device on the same bus.

3.1.17 ATA Overlapped Command

{ATA Spec} A command is an overlapped command if it is listed as part of the Overlapped or Queued Feature Set.

3.1.18 ATA PIO

{ATA Spec} For the ATA bus, PIO means that data is transferred between the device and the adapter by reading or writing a register in the device. The address of the register and the timing of the transfers are under the control of the adapter.

3.1.19 ATA Ultra-DMA

{ATA Spec} ATA Ultra-DMA is a high-speed mode of data transfer. ATA-5 defines transfer rates up to 66 MB/s, and ATA- includes a 100 MB/s rate. The Ultra protocol also defines the use of a CRC to validate that data has been correctly transferred.

3.1.20 ATAPI (AT Attachment Packet Interface) Device

{ATA Spec} An ATA device that implements the Packet Command feature set.

3.1.21 Bus Protocol

A bus protocol consists of the sequence of bus signal states, and their timings, which are required in order to transfer commands and data along a bus. There may be more than one protocol available on any one bus.

3.1.22 Command Complete

When ATA BSY = 0 and ATA RDY = 1 after a command has been delivered.

3.1.23 Cyclic Redundancy Check (CRC)

{ATA Spec} Used for the Ultra-DMA protocol to check the validity of the data that has been transferred during the last Ultra-DMA burst.

3.1.24 DMA

Direct Memory Access. A means of data transfer, between device and host memory, such that host processor intervention is not needed to accomplish the data transfer after initiation of the transfer activity.

3.1.25 Host DMA

{PCI Spec} Host DMA means that data is transferred between the host and the ATA adapter over the PCI bus, using the PCI Burst mode protocol with the adapter as master and the PCI host as target. Once initiated, the transfer requires no host Processor involvement.

3.1.26 Legacy PIO Mode

Legacy PIO Mode is an operating mode of the ADMA engine where accesses to the ATA device uses host memory or I/O instructions to access the device registers directly using ATA PIO protocols.

3.1.27 PCI

PCI is an acronym for Peripheral Component Interconnect.

3.1.28 Wintel

“Wintel” is a shorthand reference to systems which use MS Windows® on an Intel® X86 architecture.

3.2 Conventions

3.2.1 Keywords

May - A keyword that indicates flexibility of choice with no implied preference.

Reserved - A keyword indicating reserved bits, bytes, words, fields, and code values that are set aside. A reserved bit, byte, word, or field contains all zeros, and is read only.

Shall - A keyword indicating a mandatory requirement. Designers are required to implement all such mandatory requirements to ensure interoperability with other products conforming to this specification.

Should - A keyword indicating flexibility of choice with a strongly preferred alternative. Equivalent to the phrase “it is recommended”.

Vendor Specific - This term is used to describe bits, bytes, fields, and code values that are reserved for vendor specific purposes.

3.2.2 Precedence

If there is a conflict among text, figures, and tables, the precedence shall be: tables, then figures, and then text.

3.2.3 Names of Registers, Words, Bytes, Bits, Etcetera

The names of registers, words, bytes, bits, and modes begin with uppercase letters. In addition, register names are prefixed with the acronyms PCI, ADMA, or ATA, to indicate which of these register sets they belong to. For example: ATA Status Register, Status Word, Bytes 0-3, and Error Bit.

3.2.4 Signal Names

Signal functional names are shown in all uppercase letters. For example: 'CLKRUN'.

3.2.5 Signal States

A signal is 'asserted' when it is driven by an active circuit to the true state. A signal is 'de-asserted' when an active circuit drives it to the false state. A signal is 'released' when it is not actively driven to any state. Some signals have bias circuitry that pulls the signal to either a true state or a false state when no signal driver is actively asserting or de-asserting the signal.

3.2.6 Numbers

Numbers are decimal, unless specified otherwise.

Hexadecimal numbers are shown as a string of digits, 0 through 9 or A through F, followed by 'h': e.g., '1AB7h.'

Binary numbers are shown as a string of digits, 0 or 1, followed by 'b': e.g., '10110111b'.'

3.2.7 Nomenclature

B - bits
B - Bytes
G - Giga
M - Mega
N - nano
M - milli
S - second

3.2.8 Symbols

 --OR gate

4 ATA Host Adapters

The ATA interface as defined in ATA-6 describes the physical, electrical, timing, protocol and command standards required to transfer data to and from a compliant device. That standard makes certain requirements on the Host Adapter but does not define any standards for the Host. This document defines the register and physical requirements of Host Adapters. The objective is to enable Host software device drivers to be developed that can work with a Host Adapter supplied from a variety of vendors.

Host Adapters act as a bridge between the Host computers data bus and the ATA bus. Thus Host Adapters are required to meet at least two sets of standards. Host software device drivers have to be able to configure the Adapter for both the Host bus operation and the ATA bus operation. Thus this document defines, where possible, a common API (Applications Programming Interface) for those functions.

4.1 Adapter Types

The ATA interface has evolved from an original combination of a plug in Host Adapter on the IBM PC-AT. This Adapter controlled hard drives that interfaced to it using an ST506 interface. ATA drives moved the functionality of that Adapter from a plug in card into the drive. The same register set was retained and thus the most important attribute of the ATA interface was initiated, backward compatibility. Software drivers and BIOS code did not have to change.

The first ATA Host Adapters were address decoder cards plugged into the ISA bus. The cards decoded the I/O addresses of the registers in the ATA register set and connected the ISA bus to the ATA bus. All timings on the ATA bus were those of the ISA bus. As time has progressed the performance of the ATA devices have far exceeded the capabilities of the ISA bus. The majority of Host Adapters now reside on the PCI bus and the Host Adapters have become more complex involving timing and protocol conversions as a very minimum. An ATA Host Adapter can be engineered to work on just about any bus. This document is limited to the original ISA bus and the PCI bus.

4.2 Adapter Modes

4.2.1 Legacy Mode

An Adapter is in Legacy Mode when the control of the transfer is through the ATA Command and Control Block Registers. Any data transfers are via PIO mode through the Data register. The addresses of the Command or Control block are configurable in this mode.

4.2.2 Compatibility Mode

This mode is only applicable to implementations on PC systems implementing the PC architecture. An Adapter is in Compatibility when the control of the transfer is through the ATA Command and Control Block Registers and registers in the Adapter. The addresses of the Command or Control block are defined as well as the interrupt lines (IRQs). Table 1defines the four standard I/O address banks.

Channel	Command Block Registers	Control Block Register	IRQ	Alternate IRQ
Primary	IF0h-1F7h	3F6h*	14	None
Secondary	170h-177h	376h*	15	None
Tertiary	1E8h-1EFh	3EEh	11	12 or 9
Quaternary	168h-16Fh	36Eh	10	12 or 9

Table 1-Compatibility Mode Standard I/O Register Addresses

*Note to Table 1: The Control Block registers were originally defined to include a second register at 3F7h and 377h. This register address was shared with the Floppy Disk adapter on the AT architecture. The floppy adapter uses bit 7 of that register; the hard drive adapter used bits 0-6. This register is no longer used in the ATA specification.

4.2.3 PCI Native Mode

This mode is only applicable to Adapters bridging to the PCI bus. In this mode the control of the transfer is through the ATA Command and Control Block Registers and registers in the Adapter. The addresses of the Command or Control block are defined in the BAR of the Adapter and are defined by the Host software. There is only one Host interrupt line for all the channels attached to an Adapter.

4.2.4 ADMA Mode

In this mode the ATA Command and Control Block registers are not accessible to the Host. Control is exercised through a data structure held in memory and Adapter registers.

5 ISA Address Decoder Adapter

This type of Adapter is commonly called a paddle card for use in PC compatible systems. The function of the adapter is to decode the I/O addresses appropriate to the channels it controls.

5.1 Mode of Operation

5.2 Compatibility. Detection

There is no standard method to detect the presence of this type of Adapter. Software may be able to detect the presence of ATA drives by examining the ATA registers at the standard I/O addresses and thereby infer the presence of an Adapter.

5.3 Adapter Set Up

There is no standard method used to set up these Adapters. In most cases the I/O address banks are set by hard jumpers or by vendor specific registers.

5.4 ATA Bus Timings

ISA timings. No programmable timing is available.

5.5 Electrical and Physical

The electrical and physical specifications of the ATA bus are defined in the {ATA Spec}; the ISA bus characteristics are defined in the {ISA Spec}.

5.6 Registers

The compatibility register set shall be implemented.

5.7 Operation

Only PIO transfer modes are possible.

6 PCI Compatibility and Native Mode Bus Master Adapters

PCI Adapters conforming to this standard may operate in Compatibility or Native Mode. Some adapters can be configured to operate in either mode; some are fixed to one of the modes. The mode configuration may be determined from the PCI Configuration registers.

6.1 Mode of Operation

6.1.1 Compatibility Mode

Adapters operating in compatibility mode support two channels conforming to the Primary and Secondary channel address and IRQ requirements.

6.1.2 PCI Native Mode

Adapters operating in compatibility mode may support one or two channels.

6.2 Detection

The Class Code fields determine the capabilities.

6.3 Adapter Set Up

The Class Code fields determine the capabilities of an Adapter and may be used to configure the channels to Compatibility or Native mode. The PCI BARs may be used to configure and determine the I/O addresses to use to access the ATA and Adapter registers.

6.4 ATA Bus Timings

Determination of the ATA PIO timings, DMA protocols and DMA timings supported is vendor specific. Consequently configuring these attributes is vendor specific.

6.5 Electrical and Physical

The electrical and physical specifications of the ATA bus are defined in the {ATA Spec}; the PCI bus characteristics are defined in the PCI2.2 specification.

6.6 PCI Registers

The PCI Adapter implements a subset of the PCI standard type 00h configuration header register set. All registers have the standard meaning as defined in the PCI Specification, Issue 2.2. The registers with specific meanings with respect to this standard are defined below. Register contents that are otherwise defined in the PCI standard are indicates as “PCI”. Fields marked 'reserved' are all zeros and read only.

Byte Offset	Bits 31-24	Bits 23-16	Bits 15-8	Bits 7-0
00h	PCI		PCI	
04h	PCI		PCI	
08h	Class Code			PCI
0Ch	PCI	PCI	PCI	PCI
10h	Base Address 0 -- Base Address of Cmd-Block Regs, ATA Channel X			
14h	Base Address 1 -- Base Address of Control Regs, ATA Channel X			
18h	Base Address 2 -- Base Address of Cmd-Block Regs, ATA Channel Y			
1Ch	Base Address 3 -- Base Address of Control Regs, ATA Channel Y			
20h	Base Address 4 -- Base Address of ATA Bus Master Registers			
24h	Reserved			
28h	PCI			
2Ch	Subsystem ID		PCI	
30h	PCI			
34h	PCI			PCI
38h	PCI			
3Ch	PCI	PCI	PCI	Interrupt Line

6.6.1 PCI Class Code

The class code indicates that the Adapter is a mass storage controller (Base Class = 01), ATA controller (Sub-Class = 01), Native or Compatibility Mode Depending on the value in the Programming Interface Byte.

Address Offset 09h
Default Value 01010xh
Attribute Base and Sub-Class Bytes are Read Only. The attribute of the Programming Interface Byte is implementation dependant.
Size 24 bits

6.6.1.1 Programming Interface Byte

Table 2 defines the usage and values of the Programming and Interface Byte.

Bit	Description
0	Determines the mode that the primary ATA channel is operating in. Clearing this bit to zero corresponds to 'compatibility', setting the bit to one means native-PCI mode. This bit is implemented as read-only if the channel supports only one mode, or read-write if both modes are supported. For implementations that support both modes the power on and hardware reset states are vendor specific.
1	This bit indicates whether or not the primary channel has a fixed mode of operation. If this bit is cleared to zero, the mode is fixed and is determined by the (read-only) value of bit 0. If this bit is set to one, the channel supports both modes and may be set to either mode by writing bit 0.
2	Determines the mode that the secondary ATA channel is operating in. Clearing this bit to zero corresponds to 'compatibility', setting the bit to one means native-PCI mode. This bit is implemented as read-only if the channel supports only one mode, or read-write if both modes are supported. For implementations that support both modes the power on and hardware reset states are vendor specific.
3	This bit indicates whether or not the secondary channel has a fixed mode of operation. If this bit is cleared to zero, the mode is fixed and is determined by the (read-only) value of bit 0. If this bit is set to one, the channel supports both modes and may be set to either mode by writing bit 0.

Table 2: PCI Adapter bit definitions in Programming Interface byte

6.6.2 PCI Base Address Registers (BAR)

Base Address Registers 0-3 have Bit 0 hard-wired to 1 to indicate I/O space.

6.6.2.1 PCI Base Address 0
This is the base address for the command block registers for ATA Channel X.
Address Offset 10h
Default Value 000001F1h

When operating in compatibility mode any write to the BAR shall be ignored and the default value shall always be used. In compatibility mode an independent IRQ shall be provided that is connected to IRQ14. When the Adapter is disabled (using the IO Enable bit in the PCI Command register), the Adapter shall not respond to any IO addresses, and shall tri-state its IRQ connections.
Attribute Bits 31-16 may be Read Only, Bits 15-3 Read/Write, Bits 2-0 Read Only
Size 32 bits

6.6.2.2 PCI Base Address 1
This is the base address for the control register for ATA Channel X. Note that, because of the Dword alignment of PCI, the Device Control and Alternate Status Registers are at offset 02h from this base. For example, to put those registers at address 3F6h, this register shall be set to 3F4h (+ Bit 0).
Address Offset 14h
Default Value 000003F5h
When operating in compatibility mode any write to the BAR shall be ignored and the default value shall always be used.
Attribute Bits 31-16 may be Read Only, Bits 15-3 Read/Write, Bits 2-0 Read Only.
Size 32 bits

6.6.2.3 PCI Base Address 2
This is the base address for the command block registers for ATA Channel Y
Address Offset 18h
Default Value 00000171h
When operating in compatibility mode any write to the BAR shall be ignored and the default value shall always be used. In compatibility mode an independent IRQ shall be provided that is connected to IRQ15. When the Adapter is disabled (using the IO Enable bit in the PCI Command register), the Adapter shall not respond to any IO addresses, and shall tri-state its IRQ connections
Attribute Bits 31-16 may be Read Only; Bits15-3 Read/Write; Bits 2-0 Read Only.
Size 32 bits

6.6.2.4 PCI Base Address 3
If the device implements two channels this is the base address for the control registers for ATA Channel Y.
Address Offset 1Ch
Default Value 00000375h
When operating in compatibility mode any write to the BAR shall be ignored and the default value shall always be used.
Attribute Bits 31-16 may be Read Only, Bits 15-3 Read/Write, Bits 2-0 Read Only.
Size 32 bits

6.6.2.5 PCI Base Address 4
Base address of the ATA Bus Master I/O registers.
Address Offset 20h
Default Value 00000000h
Attribute Bits 31-16 may be Read Only, Bits 15-3 Read/Write, Bits 2-0 Read Only.
Size 32 bits

6.6.3 PCI Interrupt Line

The Host BIOS and O/S Drivers may use this location to store the system interrupt (IRQ) allocated to this device. The Adapter does not use this information. BIOS and O/S Drivers may use this location to store the information. When the Adapter is in compatibility mode the value in this register has no meaning since the IRQs are predetermined for each channel.
Address Offset 3Ch
Default Value 00h

6.7 ATA Bus Master Registers

The bus master IDE function uses 16 bytes of IO space. All bus master IDE IO space registers can be accessed as byte, word, or Dword quantities. The description of the 16 bytes of IO registers follows:

Offset from Base Address	Register	Register Access
00h	ATA Bus Master Command register Primary	R/W
01h	Device Specific	
02h	ATA Bus Master Status register Primary	RWC
03h	Device Specific	
04h-07h	ATA Bus Master PRD Table Address Primary	R/W
08h	ATA Bus Master Command register Secondary	R/W
09h	Device Specific	
0Ah	ATA Bus Master Status register Secondary	RWC
0Bh	Device Specific	
0Ch-0Fh	ATA Bus Master PRD Table Address Secondary	R/W

Table 3 ATA Bus Master Register Offsets

6.7.1 ATA Bus Master Command Register

Register Name: ATA Bus Master Command Register

Address Offset: Primary Channel: Base + 00h

Secondary Channel: Base + 08h

Default Value: 00h

Attribute: Read / Write

Size: 8 bits

Bit	Description
7:4	**Reserved.** Must return 0 on reads.
3	**Read or Write Control:** This bit sets the direction of the bus master transfer: when cleared to zero, PCI bus master reads are performed. When set to one, PCI bus master writes are performed. This bit shall NOT be changed when the bus master function is active.
2:1	**Reserved. Shall** return 0 on reads.
0	**Start/Stop Bus Master:** Bus master operation of the Adapter is enabled by setting this bit to one. Bus master operation begins when this bit is detected changing from a zero to a one. The Adapter shall only transfer data between the ATA device and memory only when this bit is set to one. Master operation may be halted by clearing this bit to zero. All state information is lost when a '0' is written; Master mode operation cannot be stopped and then resumed. If this bit is reset while bus master operation is still active the bus master command is aborted and data transferred from the drive may be discarded before being written to system memory. This bit is intended to be reset after the data transfer is completed, as indicated by either the Bus Master ATA Active bit or the Interrupt bit of the Bus Master ATA Status register for that ATA channel being set to one, or both.

Table 4 ATA Bus Master Command Register

6.7.2 ATA Bus Master Status Register

Register Name: Bus Master ATA Status Register

Address Offset: Primary Channel: Base + 02h

Secondary Channel: Base + 0Ah

Default Value: 00h

Attribute: Read/Write Clear

Size: 8 bits

Bit	Description
7	**Simplex only:** This read-only bit indicates whether or not both bus master channels (primary and secondary) can be operated at the same time. If the bit is a '0', then the channels operate independently and can be used at the same time. If the bit is a '1', then only one channel may be used at a time.
6	**Drive 1 DMA Capable:** This read/write bit is set by device dependent code (BIOS or device driver) to indicate that drive 1 for this channel is capable of DMA transfers, and that the controller has been initialized for optimum performance.
5	**Drive 0 DMA Capable:** This read/write bit is set by device dependent code (BIOS or device driver) to indicate that drive 0 for this channel is capable of DMA transfers, and that the controller has been initialized for optimum performance.
4:3	**Reserved.** Must return 0 on reads.
2	**Interrupt:** This bit is set to one by the rising edge of the ATA channel's interrupt line. This bit is cleared to zero when a '1' is written to it by software. Software can use this bit to determine if an ATA device has asserted its interrupt line. When this bit is read as a one, all data may have been transferred from the Device to the Host's system memory. The Adapter shall not set this bit to one until it has flushed any internal data buffers.
1	**Error:** This bit is set when the controller encounters an error in transferring data to/from memory. The exact error condition is bus specific and can be determined in a bus specific manner. This bit is cleared when a '1' is written to it by software.
0	**Bus Master IDE Active:** This bit is set to one when the Start bit is written to the ATA Bus Master Command registers. This bit is cleared to zero when the last transfer for a region is performed, where EOT for that region is set in the region descriptor. It is also cleared when the Start bit is cleared in the ATA Bus Master Command register. When this bit is read as a zero, all data transferred from the drive during the previous bus master command has been transferred to the Host's system memory, unless the bus master command was aborted.

Table 5 Bus Master ATA Status Register

6.7.3 PRD Table Pointer Register

Register Name: Descriptor Table Pointer Register

Address Offset: Primary Channel: Base + 04h

Secondary Channel: Base + 0Ch

Default Value: 00000000h

Attribute: Read / Write

Size: 32 bits

Bit	Description
31:2	Base address of Descriptor table. Corresponds to A[31:2]
1:0	Reserved

Table 6 PRD Table Pointer Register

The PRD Table must be Dword aligned. The Descriptor Table must not cross a 64K boundary in memory.

6.8 Interrupt Line Considerations

When a channel is in compatibility mode the IRQ used by the channel shall be the compatibility' IRQ. PCI interrupt lines shall not be affected by that channel's interrupt. Conversely, when the channel is in native-PCI mode the channel's interrupt shall be connected to the appropriate INTx#. Compatibility IRQs shall not be affected and if connected shall be tri-stated.

Connections of channel interrupt signals to the compatibility IRQs shall be tri-stated until the Adapter is enabled via the PCI Command register in PCI Configuration Space. The Adapter is enabled when a '1' is written to the IO enable bit (bit 0) in the Command register.

6.9 Bus Master Operation

When transferring data to or from an ATA device using an ATA DMA protocol the Adapter uses PCI Bus Master protocols to transfers the data to or from the Host's memory.

The master mode-programming interface is an extension of the standard ATA programming model. This means that devices can always be dealt with using the standard ATA programming model, with the master mode functionality used when the appropriate driver and devices are present. Master operation is designed to work with any ATA device that supports DMA transfers on the ATA bus. Devices that only work in PIO mode can be used through the standard ATA programming model.

The programming interface defines a simple scatter/gather mechanism allowing large transfer blocks to be scattered to or gathered from memory. This cuts down on the number of interrupts to and interactions with the CPU.

Master ATA Adapters shall default to Mode 0 Multiword DMA timings to ensure operation with DMA capable ATA devices without the need for Adapter vendor-specific code to initialize Adapter-specific timing parameters.

6.9.1 Physical Region Descriptor Table

Before the Adapter starts a master transfer it is given a pointer to a Physical Region Descriptor Table (PRD Table). This table contains some number of Physical Region Descriptors (PRDs), which describe areas of memory that are involved in the data transfer. The descriptor table must be aligned on a 4-byte boundary and the table cannot cross a 64K boundary in memory.

6.9.2 Physical Region Descriptor

The physical memory region to be transferred is described by a Physical Region Descriptor (PRD). The data transfer will proceed until all regions described by the Prods in the table have been transferred.

Each PRD entry is 8 bytes in length. The first 4 bytes specify the byte address of a physical memory region. The next two bytes specify the count of the region in bytes (64K byte limit per region). A value of zero in these two bytes indicates 64K. Bit 7 of the last byte indicates the end of the table; bus master operation terminates when the last descriptor has been retired.

	Byte 3		Byte 2	Byte 1	Byte 0	
Dword 0	Memory Region Physical Base Address [32:1]					0
Dword 1	EOT	Reserved		Byte Count [15:1]		0

Figure 1 Physical Region Descriptor Table Entry

The memory region specified by the descriptor is further restricted such that the region cannot straddle a 64K boundary. The sum of the descriptor byte counts must be equal to, or greater than the size of the disk transfer request. If greater than, then the driver must terminate the Bus Master transaction (by resetting bit zero of the ATA Bus Master Command register to zero) when the drive issues an interrupt to signal transfer completion.

6.9.3 Standard Programming Sequence

To initiate a bus master transfer between memory and an ATA DMA device, the following steps are required:

1. Software prepares a PRD Table in system memory. Each PRD is 8 bytes long and consists of an address pointer to the starting address and the transfer count of the memory buffer to be transferred. In any given PRD Table, two consecutive Prods are offset by 8-bytes and are aligned on a 4-byte boundary.
2. Software provides the starting address of the PRD Table by loading the PRD Table Pointer Register. Setting of the Read/Write Control bit specifies the direction of the data transfer. Clearing the Interrupt and Error bits in the ATA Bus Master Status register to zero readies the Adapter for a data transfer.
3. Engage the bus master function by writing a '1' to the Start bit in the ATA Bus Master Command Register for the appropriate channel.
4. Software issues the appropriate DMA transfer command to the disk device.
5. The controller transfers data to/from memory responding to DMA requests from the ATA device.
6. At the end of the transfer the ATA device signals an interrupt.
7. For transfers from the Device to the Host the Adapter shall first flush any internal data buffers before asserting the Host interrupt signal.

8. In response to the interrupt, software resets the Start/Stop bit in the ATA Bus Master Command register. The software then reads the Adapter status and then the Device status to determine if the transfer completed successfully.

6.9.4 ATA Bus Master Status Register Bit Interpretation

The table below gives a description of how to interpret the Interrupt and Active bits in the Controller status register after a DMA transfer has been started.

Interrupt	Active	Description:
0	1	DMA transfer is in progress and the ATA device has not asserted an interrupt.
1	0	The ATA device asserted the interrupt signal and the Adapter exhausted the Prods. This is the normal completion case where the size of the physical memory regions was equal to the ATA device transfer size.
1	1	The ATA device asserted the interrupt signal but the Adapter has not reached the end of the physical memory regions. This is a valid completion case where the size of the physical memory regions was larger than the ATA device transfer size.
0	0	This bit combination signals an error condition. If the Error bit in the ATA Bus Master Status register is set to one, then the Adapter has some problem transferring data to/from memory. Specifics of the error have to be determined using bus-specific information. If the Error bit is not set, then the Prod's specified a smaller size than the ATA transfer size.

Table 7 ATA Bus Master Status Register Bits

6.9.5 Error Conditions

If the controller encounters an error while doing the bus master transfers it will stop the transfer (i.e. reset the Active bit in the ATA Bus Master Command register) and set the ERROR bit in the ATA Bus Master Status register. The controller does not generate an interrupt when this happens. The device driver may use device specific information (e.g.; PCI Configuration Space Status register) to determine what caused the error.

7 ADMA Mode - General Description

7.1 Background

The performance of ATA devices has increased dramatically over the last few years, including the introduction of the Overlapped and Queued Feature Sets. However, systems using these faster devices have not shown all of the expected benefits. This limited performance improvement can be traced to the design and implementation of the ATA host adapters currently in use. The problem in Wintel systems is amplified by the design of the standard ATA driver software used in Windows® operating systems. These drivers effectively treat ATA transfers as a single-threaded entity, with correspondingly long latencies to fully service interrupts. The ADMA engine is designed to address these problems and add features that make it, and ATA devices, suitable for true multi-threading applications. These include network server and streaming audiovisual applications.

7.2 The ADMA

The objective of the ADMA design is to drastically increase the performance of systems that use ATA devices.

To fully optimize the system throughput, the ADMA implements a command chaining technique to de-couple the host command sequence from the channel execution. This decoupling is accomplished by having the traditional host software-to-device I/O negotiation executed in hardware. This allows true multitasking and the ability to effectively exploit the Overlapped and Queued Feature Sets.

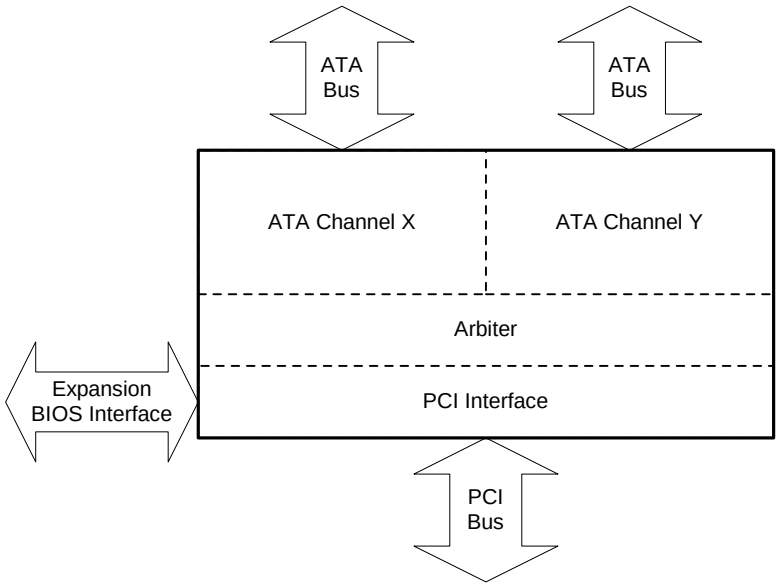

Figure 2 – ADMA Outline Block Diagram

8 ADMA Benefits: Host Overhead Reduction

There are a number of factors that contribute to the time taken by the host to service an I/O request. This section examines the requirements of the ATA protocol and of Windows® drivers, both of which contribute to the time taken to service an I/O request. ADMA command chaining is shown to drastically reduce host overhead.

8.1 Command Chain Solution Within ADMA Hardware

In conventional host hardware, the software performs read/write sequences using I/O instructions. For any one transfer, the PCI bus utilization is determined by the I/O timing of the device. Traditionally, when the command is complete, the ATA device asserts an interrupt. The host device driver is activated by the interrupt, and then reads the ATA Status Register to determine if the transfer was successful and to clear the pending interrupt. Depending on system load, the period between the interrupt being asserted and the device driver being activated can be milliseconds. The relationship between the host software (device driver) and the ATA device is *closely coupled*, with each I/O Request requiring a separate service from the device driver before it can begin execution of the next.

The principle objective of ADMA command chaining is to allow the host software and the ATA device to be *loosely coupled*. To do this, the host software builds up a list of I/O Requests for the ADMA to complete. These are stored in system memory using a “command chain” structure. The ADMA hardware *independently* reads these requests from the command chain at DMA burst speeds and passes them to the ATA device. When the ATA device completes a command, it asserts an interrupt. The ADMA engine updates the command chain with status information, and then asserts an interrupt to inform the host. The ADMA engine immediately proceeds to the next I/O Request *without waiting*.

For example, under Windows® a device driver receives an I/O Request from the I/O Manager and puts it onto the command chain. If the I/O Manager passes in another I/O Request, it too is put onto the chain. I/O Requests can be added onto the chain until the device driver's workspace is exhausted. Concurrent with host I/O activity mentioned above, the ADMA is reading the command chain and executing requests. After each request is complete, the ADMA asserts the PCI INTA signal to tell the host the I/O Request is completed. At this time, the device driver completes the processing of *one or more* I/O Requests. The ADMA does not have to wait for a previous I/O Request to be completed, but can immediately continue with the next request in the chain.

With ADMA command chaining, the overhead associated with writing commands to the device is greatly reduced. This is directly attributable to the fact that the delay imposed between the execution of one I/O request and the initiation of the next is no longer dependant upon the interrupt latency (that interval occurring between the time a device signals an interrupt, to the time the interrupt is handled by the device driver) of the system in which the device is running.

The ADMA command chain is a linked-list of command parameters known as a Command Parameter Block (CPB). The CPB contains all the information needed to control both the ADMA and the ATA device. Additionally, the CPB points to a second linked-list that defines the scatter-gather relationship of the memory buffers to be used for data transfer. This second chain is termed a Physical Region Descriptor (PRD).

8.2 ADMA Enhancement of ATA Queued Commands

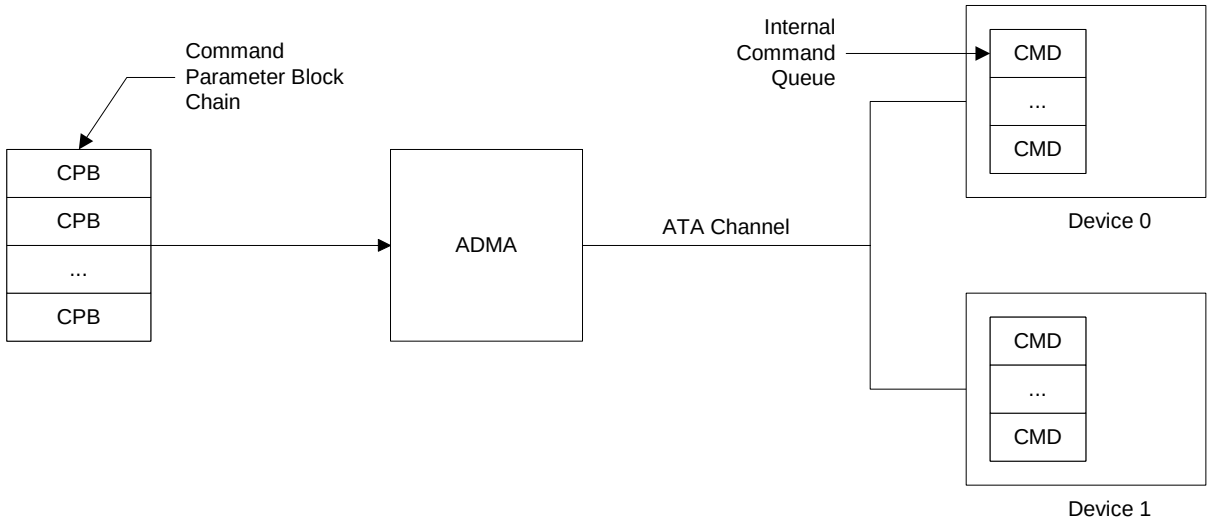

Figure 3 – ADMA Assisted Command Queuing

8.2.1 Command Delivery

Queued commands allow an ATA device to improve its performance by storing a series of I/O requests in an internal command queue. I/O Requests may then be reordered to optimize response time. For example, a hard disk may reorganize requests to streamline disk accesses, effectively reducing overall seek time. After a device accepts a queued command, there may be a period of time before it is ready to transfer data. During this time, more commands can be delivered to the device in order to allow it to optimize its performance. Conventionally, the delivery of more commands to the device requires a great deal of processor overhead; thus, only a few queueable commands could be delivered to the device over a set period of time, resulting in a small internal device command queue. Using the command chaining method discussed in Section 8.1, the ADMA is capable of delivering more commands to the device over a shorter period of time, allowing a larger internal command queue to be built by the device. See Figure 3. In this way the ADMA both reduces processor overhead and substantially increases the number of commands that can be loaded into an internal command queue in the ATA device.

8.2.2 Auto-Polling

Queued commands can be sent to two devices on a single channel, but only the device that is currently selected may inform the host that it is ready to be serviced. In conventional systems, the host software has to continually alternate selection of the devices (poll) to watch for either device to become ready. This polling process imposes a substantial host CPU overhead. The ADMA hardware, independent of the host CPU, automatically polls each device (Auto-Polling) and only performs the transfer when a device is ready. This design eliminates any polling overhead in the host.

9 ADMA Functional Overview

The ADMA controls two ATA Channels. These channels are termed Channel X and Channel Y, and are functionally identical. Figure 4 represents a simplified view of the major functional blocks within the ADMA. There are four interfaces to the ADMA engine: the PCI bus interface, the Extension BIOS interface, and two ATA/ATAPI channels. The following sections discuss the major functional blocks within the ADMA hardware, and introduce the principal features supplied.

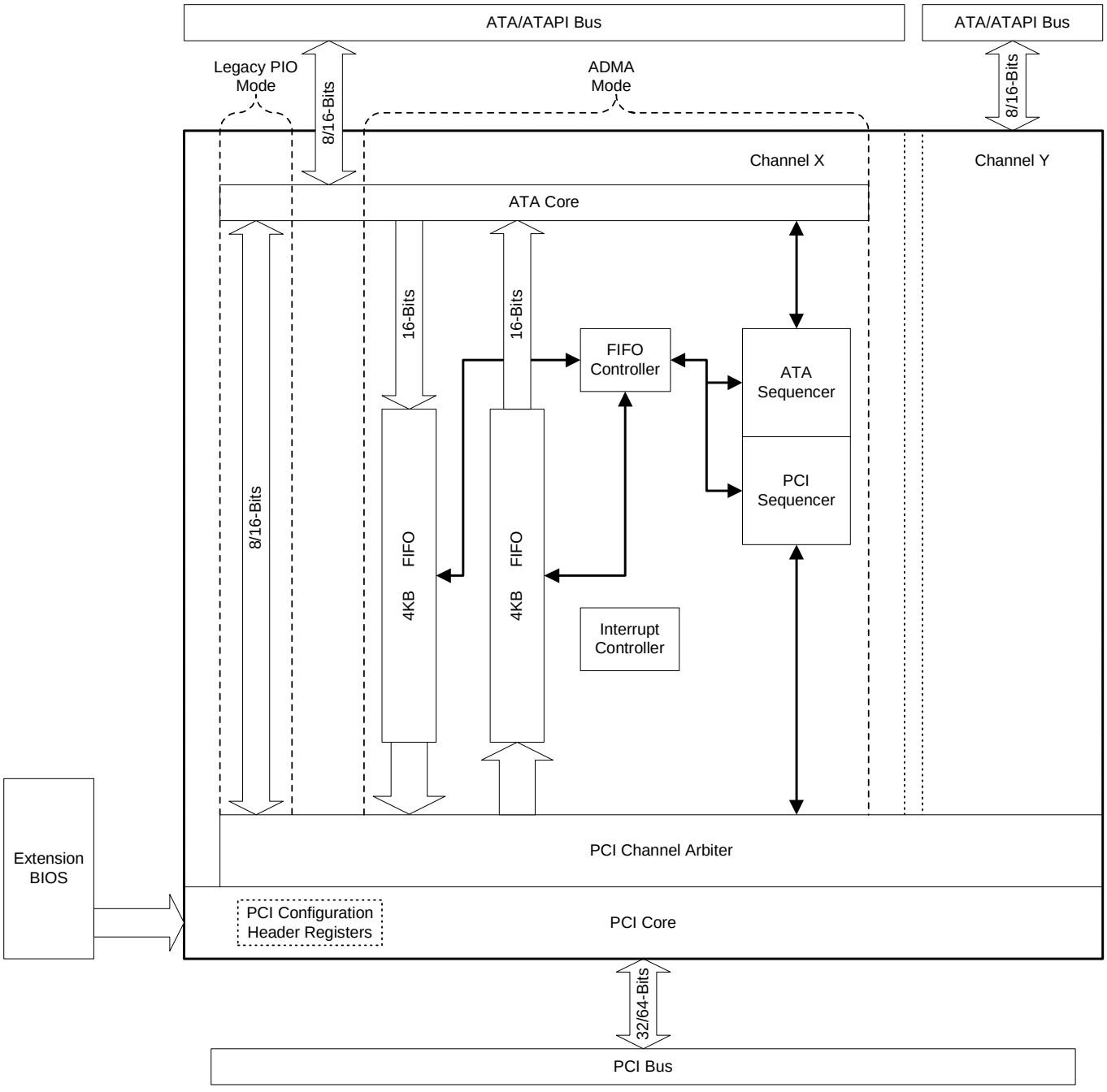

Figure 4 – ADMA Block Diagram

9.1 ADMA Block Diagram

9.1.1 Modes of Operation

The ADMA engine works in two modes, termed *Legacy PIO* and *ADMA*. In Legacy PIO Mode, the host software directly writes/reads the ATA registers. The only function performed by the ADMA engine is to obey the Programmed I/O (PIO) timing rules for the current PIO mode. In this mode, the ATA interrupts are directly

mapped onto the PCI interrupt. In ADMA Mode, the ADMA engine controls all aspects of the ATA protocols including interception of the ATA interrupt as appropriate.

9.1.2 PCI Bus

The PCI bus conforms to the PCI 2.2 specification, the {PCI PMS}, and the {PCI Hot Plug} requirements. The ADMA contains configuration header registers consistent with its implementation as a single PCI function device, and thus drives a single interrupt line (the PCI INTA signal), see Section 9.1.10.

9.1.3 PCI Core

The PCI core is responsible for implementing the PCI protocols for both master and target operations. The PCI configuration header registers contain one Memory Mapped Base Address Register (BAR) associated with the ADMA Registers, two I/O BARs for each ATA channel, and an Expansion ROM BAR associated with the BIOS.

9.1.4 PCI Channel Arbiter

The Channel Arbiter interleaves operations from both channel X and Y, making the two channels functionally independent.

9.1.5 FIFOs and FIFO Controller

Each channel contains two 4KB FIFOs, input and output, used to buffer data during transfers. These FIFOs are a power of two Quad-Words (Qwords) long and are controlled by two host programmable registers: the ADMA FIFO Input Threshold (FITR) and ADMA FIFO Output Threshold (FOTR). These registers are used by the FIFO Controller to indicate the burst size to/from the PCI core. The ADMA FIFO Input Threshold controls the requested PCI data transfer burst length from the ADMA to the host, while the ADMA FIFO Output Threshold controls the requested PCI data transfer burst length from the host to the ADMA. Adjusting the FIFO thresholds controls the length of the transaction.

The FIFO interface to the ATA Channel is 16-bits wide in both directions.

9.1.6 PCI and ATA Sequencer

Each channel within the ADMA has its own sequencers (PCI and ATA) that operate completely independently of the other channel.

The PCI Sequencer is responsible for reading the current ATA command and data using PCI master mode bursts.

The ATA Sequencer handles the ATA/ATAPI protocols.

9.1.7 ATA Core

The ATA standard defines various PIO transfer rates, the theoretical maximum of which is 16.666MB/s. The ATA PIO protocol is automated in the ADMA, so that PIO is used to transfer data to/from the device while using PCI DMA burst mode from/to the host. This method of operation is termed "DMA-Assisted PIO mode". The ADMA uses DMA-Assisted PIO mode for devices not implementing Ultra-DMA.

In Legacy PIO Mode, the ADMA supports data transfer to/from the ATA bus using PIO protocols.

In Full-DMA mode, the ADMA uses Ultra-DMA between the device and the ADMA and uses PCI Burst mode DMA between the ADMA and the host.

9.1.8 ATA/ATAPI Channel

The ATA/ATAPI channel conforms to the {ATA Spec}.

9.1.9 Extension BIOS

The base address of the extension BIOS is in the PCI Expansion ROM Base Address Register.

9.1.10 Interrupt Controller

The ADMA Interrupt Controller generates the ADMA interrupt signal shown in Figure 5, and processes three different sources of interrupt:

1. ADMA Interrupts. The ADMA Interrupt signal is asserted by the ADMA to inform the host software of certain important events.
2. Unsolicited Interrupts. The ATA Unsolicited Interrupt (UIRQ) signal allows the ADMA to provide support for insertion and removal request notification of removable devices.

3. Legacy Interrupts. When in ATA Legacy PIO Mode, the ATA Legacy Interrupt is the ATA INTRQ signal coming from the ATA/ATAPI bus.

The above signals from each channel are OR-ed to generate the Channel X Interrupt and the Channel Y Interrupt. These signals are OR-ed in turn to generate the PCI INTA signal.

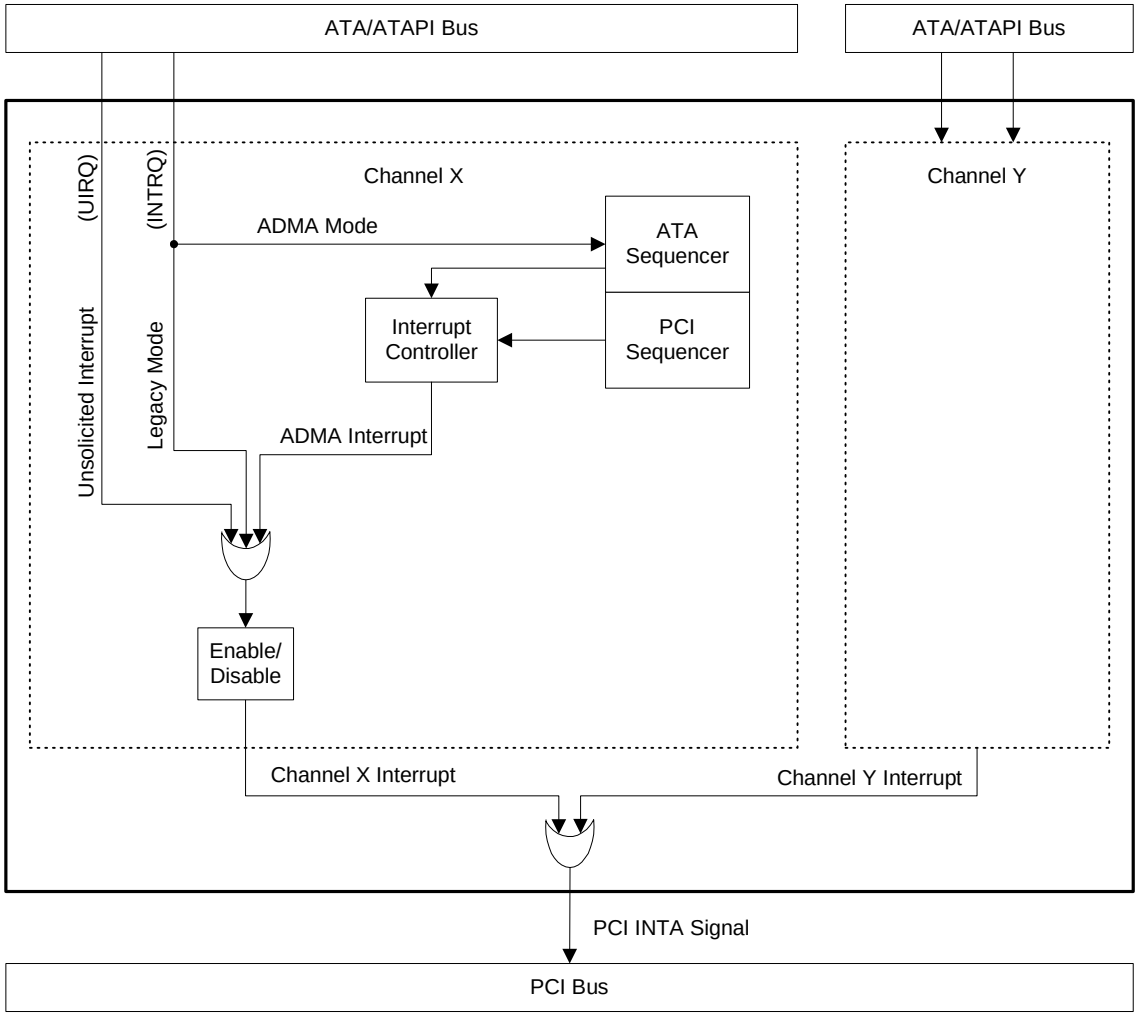

Figure 5 – ADMA Interrupts

9.2 ADMA Features Overview

The features of the ADMA engine are designed to provide access to the ATA registers using PIO reads and writes. Many new features drastically improve performance as well as functionality. This section briefly introduces these features.

9.2.1 Single Stepping and Continuous DMA

The ADMA is available in two versions, Single Stepping DMA and Continuous DMA. The single stepping DMA version processes one command at a time, interrupting the host after each command is complete. The continuous DMA version allows multiple commands to be put into a chain. These commands are processed without host intervention.

9.2.2 Accessing ATA Registers

The ATA device's Command and Control Registers are accessible through the ADMA by either I/O or memory mapped registers. The control of the ADMA engine is through a set of memory mapped registers.

9.2.3 ADMA Control Operating Modes

Two modes of operation are implemented: Legacy PIO Mode, and ADMA Mode.

9.2.3.1 Legacy PIO Mode

Legacy PIO Mode is the power-on and reset default. In this mode, the ATA adapter acts as an address decoder for the host. All reads and writes are performed using host I/O or host Memory instructions. The only function performed by the ADMA is to control the signal timings of the ATA bus using the ATA core, and to respond to PCI signals. In this mode, all data transfers use the PIO protocols, and ATA bus interrupts are directly mapped onto the PCI INTA signal. Note that both channels X and Y use the single PCI INTA signal as shown in Figure 5.

9.2.3.2 Automatic DMA Mode

The ADMA autonomously follows a command chain in memory as described in Section 8.1. In this mode, data transfers can be either Ultra-DMA or DMA-assisted PIO.

In ADMA Mode, the ATA legacy registers are not available. In this mode, any access to the ATA Status (or ATA Alternate Status) register returns a value with bit 7 (BSY) set to one. Any read to other registers returns an indeterminate value. Any write to an ATA register is ignored.

9.2.4 Frequently Used ADMA Registers

The ADMA Registers for both channels X and Y are addressed through BAR 4, which is offset 20h in the PCI configuration header registers.

9.2.4.1 ADMA Control (ADMCTL)

The ADMA Control Register contains bits to control the mode and flow of operation.

9.2.4.2 ADMA Status (ADMSTAT)

The ADMA Status Register contains bits that indicate the current state of the process and interrupt reason.

9.2.4.3 CPB Search Count (CCNT)

The ADMA uses this value to determine when to stop processing CPBs. This value is usually set to the number of entries in the CPB chain.

9.2.4.4 Current CPB Address (CCPB)

The Current CPB Address Register contains the 32-bit address of the CPB currently being processed.

9.2.4.5 Next CPB Address (NCPB)

The ADMA Next CPB Address Register contains the 32-bit address of the next CPB to be processed by the ADMA. The host software shall initialize the contents of this register before entering ADMA Mode. The ADMA updates this register as it progresses through the CPB chain, except when accessing the CPB Lookup Table.

9.2.4.6 CPB Lookup Table Address Register (CPBLAR)

The CPB Lookup Address Register contains the 32-bit address of the base of the CPB Lookup Table. If the host software is using overlapped/queued commands, it shall initialize the contents of this register before entering ADMA Mode. The ADMA uses this address to locate the correct CPB when servicing a released overlapped/queued ATA Command.

9.2.4.7 ADMA FIFO Input Threshold Register (FITR)

FITR controls the requested PCI data transfer burst length from the ADMA to the host.

9.2.4.8 ADMA FIFO Output Threshold Register (FOTR)

FOTR controls the requested PCI data transfer burst length from the host to the ADMA.

9.2.5 ADMA Data Structures

The ADMA requires the following data structures to control and manage ATA devices when in ADMA Mode. These data structures shall be physically located in memory with each entry of the data structure in physically contiguous space. All addresses shall be Qword aligned.

9.2.5.1 CPB Chain

The CPB chain is a circular linked-list of CPB entries. A CPB is a data structure used during ADMA Mode to store parameters that control the ADMA engine, and define ATA command(s) for the ATA device. A chain of CPBs is created in memory as a circular linked-list with each CPB pointing to the next CPB. Within each CPB, there is a pointer to a PRD chain. A PRD chain defines the memory locations where data is to be written to/read from. Figure 6 illustrates the CPB structure and PRD relationship.

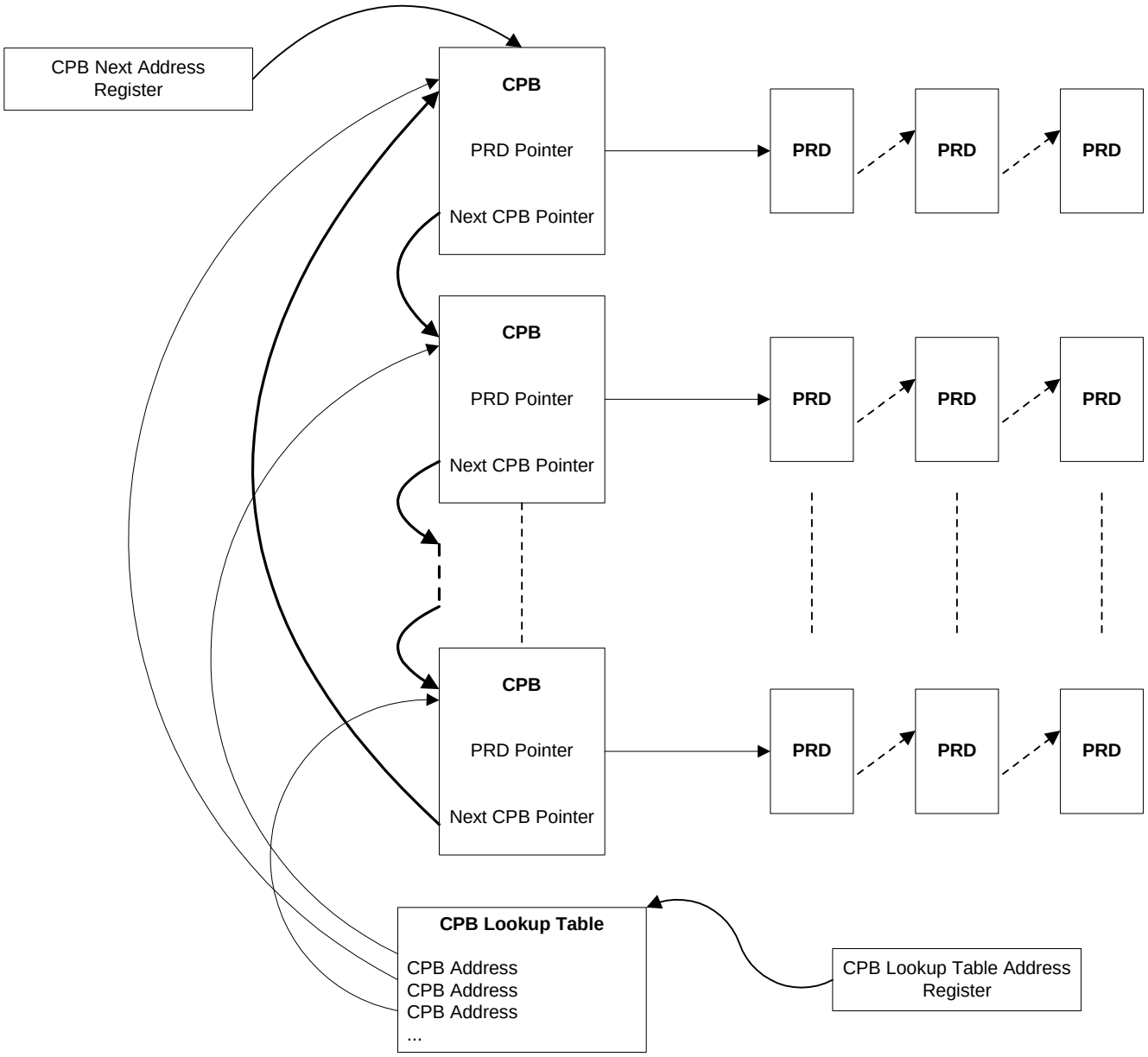

Figure 6 - ADMA Data Structures

An individual CPB can be in one of four States: Not-Valid, Valid-Waiting, Valid-Processing or Released. A CPB that is Not-Valid is under the control of the host. A CPB that is Valid-Waiting, Valid-Processing, or Released is under the control of the ADMA and should not be updated by the host, except for the cVLD bit (see Section 12.3).

9.2.5.2 PRD Chain

A PRD chain is a linked-list of one or more PRD entries. Each entry contains the physical location to be used as the source or destination of the current transfer, and the transfer length. The transfer location may be a contiguous memory area or an I/O address. Each entry also contains a pointer to the next entry, as well as values to control the method and mode of the transfer.

9.2.5.3 CPB Lookup Table

The host constructs a CPB Lookup Table in memory for use in overlapped/queued operation. It is pointed to by the CPB Lookup Address Register, which is initialized by the host software. Each entry in the table is a Qword holding the physical address of a CPB. Figure 6 illustrates the CPB Lookup Table.

9.2.5.4 Data Structure Initialization

A valid PRD chain shall be constructed. A PRD chain defines the memory locations where data is to be written to/read from. A valid CPB chain shall be constructed consisting of one or more CPB entries with the Next CPB field pointing to the physical memory address of the next CPB in the chain (the Next CPB field in a chain of one CPB would point to itself). Each CPB shall be initialized to point to the head of a PRD chain. All CPBs shall be set to Not-Valid. The CPB Lookup Table shall be constructed. The address of the first CPB shall be written into the ADMA Next CPB Address Register, and the base of the CPB Lookup Table shall be written into the ADMA Lookup Table Address Register.

9.2.6 ADMA Engine Initialization

Before entering ADMA Mode, the host writes a CPB address into the ADMA Next CPB Address Register, and writes into the CPB Search Count Register the number of CPBs to scan before stopping. The value in the CPB Search Count Register is used by the ADMA to refresh an internal counter. This internal counter is refreshed each time the host indicates that the ADMA should examine the CPB chain. The ADMA decrements this internal count each time a CPB is encountered. When this internal counter reaches zero, the ADMA stops fetching CPB entries. The value set in the CPB Search Count Register will normally relate to the number of CPBs in the CPB chain.

9.2.7 Time-outs

The ADMA does not automatically time out an ATA command. The host software is responsible for timing out events. Status information in the CPB chain and the ADMA Registers is available for host software to be able to determine the status of the ADMA. Host software is able to pause the ADMA in the event of a system time-out, revert to Legacy PIO Mode, and directly address the device registers.

9.2.8 Non-Queued Operation

The host software assembles CPB entries in the host's memory and indicates that the ADMA should examine the CPB chain. Starting from the current value in the ADMA Next CPB Address Register, the ADMA Sequencer reads the current CPB using a PCI master mode burst.

During the CPB read:

1. The contents of the CPB ATA command block are transferred into the ADMA FIFO
2. The Next CPB Address Register is updated from the current CPB to point to the next CPB in the chain
3. The ADMA stores the start address of the PRD chain

If the CPB is ready to be processed, the ATA device registers are written to initiate the transfer. If the CPB is not ready to be processed, the next CPB is read until a valid entry is found or the ADMA internal CPB counter has decremented to zero.

If the CPB involves a data transfer, the ADMA reads the first PRD entry, using a PCI master mode burst. The data address and transfer count in this entry are used by the ADMA to control the data transfer. The control information in this entry is used by the ADMA to manage the transfer on the ATA Bus. When all the data has been transferred as indicated by the PRD count, the ADMA sequencer accesses the next PRD entry from the address located in the current PRD. This process is repeated until the total transfer is complete.

If the ATA data transfer is via PIO, the ADMA monitors the ATA Status Register and the ATA INTRQ signal, to determine when the device is ready to transfer data.

If the transfer is via Ultra-DMA, the ATA device indicates that it is ready, by asserting the ATA signal DMARQ.

At the conclusion of a CPB, the ADMA updates the CPB with status information. The PCI INTA signal is then asserted if the CPB indicates a request for an interrupt on completion of the CPB, or if an ATA error occurs.

9.2.9 Queued Operation

The overlapped/queued protocols differ for ATA and ATAPI devices. Only certain commands can be overlapped or queued. On each such command, the CPB indicates that it can be queued. After writing the command to the ATA device, the ADMA waits for the ATA INTRQ or DMARQ signal to be asserted, or ATA BSY to be cleared to zero.

If an ATA device is ready to transfer data, it asserts DMARQ. If the device is not ready to transfer data for the current command, it may release the ATA bus, by asserting the ATA INTRQ signal with the ATA Release Bit (REL) set to one. In either case, the ATA Service Bit (SERV) may be set to one, to indicate that the device is ready to transfer data for a previously queued command.

If the ATA INTRQ signal is asserted and ATA REL is set to one and ATA SERV is cleared to zero, then the ADMA sets the current CPB to the Released State and steps on to the next CPB without asserting the PCI INTA signal. The next valid CPB is executed, by writing the command to the specified ATA device, thereby creating a queue of commands in one or both ATA devices.

If ATA SERV is set to one, the ADMA reads the ATA DEV Bit to determine the current device, writes an ATA Service command to the device, and then polls the ATA status Register. When ATA DRQ is set to one, the ADMA reads the TAG from the ATA device and combines the TAG, DEV, and CPBLAR to fetch the Released CPB's address from the CPB Lookup Table. The ADMA then performs the data transfer as in the non-queued case.

The ADMA alternately selects each device (auto-poll) when the ATA bus is in the IDLE State and auto-polling is enabled. This enables a device to assert the ATA INTRQ signal to indicate that it requires service. The ADMA stops the auto-poll sequence if the ATA INTRQ signal is asserted.

9.2.10 Enhanced Data Integrity

Ultra-DMA mode transfers include the use of a Cyclic Redundancy Check (CRC). The CRC is calculated over the entire block of data transferred. A CRC was introduced into the Ultra-DMA protocol to increase the integrity of the data on the ATA cable. The effectiveness of a CRC reduces as the length of the transfer increases. The ADMA has the ability to break each transfer into smaller units by terminating the burst after transfer of a host-specified number of Sectors. The maximum number of Sectors in each burst is defined in the PRD for that block of data.

10 PCI Registers

10.1 PCI Configuration Header Registers

All ADMA PCI registers have the standard meaning as defined in the PCI Specification, Issue 2.2. The ADMA implements a subset of the standard type 00h configuration header register set. The implemented registers, and device-specific values, are described below. Fields marked 'Reserved' contain all zeros and are read only.

Bits 31-24	Bits 23-16	Bits 15-8	Bits 7-0	Byte Offset
PCI Device ID		PCI Vendor ID		00h
PCI Status Register		PCI Command Register		04h
PCI Class Code			PCI IC Revision	08h
Reserved	PCI Header Type	PCI Latency Timer	PCI Cache Line Size	0Ch
Base Address 0 – Base Address of Command-Block Registers, ATA Channel X				10h
Base Address 1 – Base Address of Control Registers, ATA Channel X				14h
Base Address 2 – Base Address of Command-Block Registers, ATA Channel Y				18h
Base Address 3 – Base Address of Control Registers, ATA Channel Y				1Ch
Base Address 4/5 – Base Address of Memory Mapped ATA Channel and ADMA Registers				20h-27h
Reserved				28h
PCI Subsystem ID		PCI Subsystem Vendor ID		2Ch
PCI Expansion ROM Base Address				30h
Reserved			PCI Capability Ptr.	34h
Reserved				38h
PCI Max. Latency	PCI Min. Grant	PCI Interrupt Pin	PCI Interrupt Line	3Ch
Vendor Specific				…
Power Management Capability		Ptr. to Nxt Capability	Capability ID	50h
Data Register	Bridge Support Ext.	Power Management Control/Status		54h
Vendor Specific				58-FFh

Table 8 – PCI Configuration Space Header Registers

10.1.1 PCI Vendor ID

Address offset 00h
Value xxxxh (Pacific Digital Corp)
Attribute Read Only
Size 16 bits

10.1.2 PCI Device ID

Indicates that the device conforms to the ADMA specification (this document).

Address offset 02h
Value 1841h
Attribute Read Only
Size 16 bits

10.1.3 PCI Command Register

The value in this register is set by the host to enable various PCI functions. Default on reset is everything disabled.

Address Offset 04h
Default Value 0000h
Attribute Read/Write
Size 16 bits

Bit	Attribute	Description
0	R/W	Target I/O enable. A value of zero disables Base Address Registers 0 – 4.
1	R/W	Memory Space enable. A value of zero disables the Expansion ROM Base Address Register and the Memory Base Address Register.
2	R/W	Master enable. A value of zero disables the Master mode function of the ADMA.
3	R	Reserved.
4	R/W	Memory Write and Invalidate Enable. A value of zero disables the function in the ADMA.
5	R	Reserved.
6	R/W	Parity check enable. A value of zero causes parity errors to be ignored.
15:7	R	Reserved.

Table 9 – PCI Command Register

10.1.4 PCI Status Register

Provides status information related to PCI bus events. Bits indicated as “clear” are cleared by writing a one to that bit position.

Address Offset 06h
Default Value See Table Below
Attribute Read Only/Clear
Size 16 bits

Bit	Description	Default	Fixed/ Clear
3:0	Reserved	0	F
4	Capabilities Enable. Set to one to indicate Capabilities are enabled.	1	F
5	Set to one to indicate 66MHz-Capable	1	F
6	Reserved	0	F
7	Not Fast Back-to-Back Capable. Cleared to zero.	0	F
8	Set when, in Master mode, a Data Parity error is detected, and bit 6 of the PCI command Register is set to one.	0	C
10:9	DEVSEL timing set to medium speed	01b	F
11	Signaled target abort	0	C
12	Received target abort	0	C
13	Received master abort	0	C
14	Reserved	0	F
15	Detected parity error	0	C

Table 10 – PCI Status Register

10.1.5 PCI IC Revision

Indicates that it conforms to revision 4 of this specification series. (I.e., revision 4 of this document.)

Address Offset 08h
Default Value 4xh
Attribute Read Only
Size 8 bits

4xh indicates that this device conforms to this specification. 40h indicates that this device does not support ATAPI devices in ADMA Mode. 41h indicates that this device does support ATAPI devices in ADMA Mode.

10.1.6 PCI Class Code

The class code indicates that the ADMA is a mass storage controller (01), ATA controller (05), Bus Master Command Chain PCI Native mode only. Devices with a programming interface code of 20h operate as though aPSE is permanently set to one (see Sections 13 and A.9).

Address Offset 09h
Default Value 010520h (Single Stepping Controller)
010530h (Continuous Operation Controller)
Attribute Read Only
Size 24 bits

Bits	Offset	Description	Value
15:8	09h	Programming Interface Code	20h – Single Stepping 30h – Continuous Operation
23:16	0Ah	Sub-class Code	05h – ATA
31:24	0Bh	Base-Class Code	01h – Mass Storage

Table 11 – PCI Class Code

10.1.7 PCI Cache Line Size

In Master mode, the Cache Line Size Register is used to determine the PCI command appropriate for the burst. The commands implemented are Memory Read, Memory Read Line, Memory Read Multiple, Memory Write, and Memory Write Invalidate. Set by the host BIOS or Operating System.

Address Offset 0Ch
Default Value 00h
Attribute Read/Write
Size 8 bits

10.1.8 PCI Latency Timer

The host writes a value (in PCI clocks) that is decremented during a master mode transfer. If the host removes the PCI GNTn signal before this value expires, the ADMA can continue. This register is set by the host BIOS or Operating System.

Address Offset 0Dh
Default Value 40h
Attribute Read/Write
Size 8 bits

10.1.9 PCI Header Type

Indicates that the ADMA is a single-function device.

Address Offset 0Eh
Default Value 00h
Attribute Read Only
Size 8 bits

10.1.10 PCI Base Address Registers (BAR)

Base Address Registers 0-3 have bit 0 hard-wired to one to indicate I/O space, and bits 16-31 hard-wired to zero. Full address decoding is still implemented. BARs 4-5 have bit 0 hard-wired to zero to indicate memory address space.

10.1.10.1 PCI Base Address 0

This is the base address for the command block registers for ATA Channel X.

Address Offset 10h
Default Value 000001F1h
Attribute Bits 31-16 Read Only, bits 15-3 Read/Write, bits 2-0 Read Only.
Size 32 bits

10.1.10.2 PCI Base Address 1

This is the base address for the Control Registers for ATA Channel X. Note that, because of the Dword alignment of PCI, the device Control and Alternate Status Registers are at offset 06h from this base.

Address Offset	14h
Default Value	000003F1h
Attribute	Bits 31-16 Read Only, bits 15-3 Read/Write, bits 2-0 Read Only.
Size	32 bits

10.1.10.3 PCI Base Address 2

This is the base address for the command block registers for ATA Channel Y.

Address Offset	18h
Default Value	00000171h
Attribute	Bits 31-16 Read Only; bits15-3 Read/Write, bits 2-0 Read Only.
Size	32 bits

10.1.10.4 PCI Base Address 3

This is the base address for the Control Registers for ATA Channel Y.

Address Offset	1Ch
Default Value	00000371h
Attribute	Bits 31-16 Read Only, bits 15-3 Read/Write, bits 2-0 Read Only.
Size	32 bits

10.1.10.5 PCI Base Address 4 and 5

This is the base address for the 64-bit Memory Mapped ATA Channel and ADMA registers.

Address Offset	20h
Default Value	0000000000000004h
Attribute	Bits 31-10 Read/Write; bits 9-0 Read Only.
Size	64 bits

10.1.11 PCI Subsystem Vendor ID

The PCI subsystem vendor ID indicates the vendor of the adapter.

Address Offset	2Ch
Default Value	xxxxh
Attribute	Read Only
Size	16 bits

10.1.12 PCI Subsystem ID

The PCI subsystem ID indicates the adapter implementation.

Address Offset	2Eh
Default Value	xxxxh
Attribute	Read Only
Size	16 bits

10.1.13 PCI Expansion ROM Base Address

Address Offset	30h
Default Value	If ROM is present 000E0000h, if no ROM is present 00000000h.
Attribute	Bits 31-17 Read/Write; bits 16-01 Read Only; bit 0 Read/Write.
Size	32 bits

10.1.14 PCI Capability Pointer

The PCI Capability Pointer points to a linked list of capabilities (i.e., the Power Management Registers).

Address Offset	34h
Default Value	50h
Attribute	Read Only
Size	8 bits

10.1.15 PCI Interrupt Line

Note: the host BIOS loads the system interrupt (IRQ) allocated to this device. This Register is not used by the ADMA. It may be used by the system BIOS and host Operating System Drivers as a location to store the IRQ being used.

Address Offset 3Ch
Default Value 00h
Attribute Read/Write
Size 8 bits

10.1.16 PCI Interrupt Pin

The PCI interrupt pin defaults to 01h, indicating that the PCI INTA signal is used.

Address Offset 3Dh
Default Value 01h
Attribute Read Only
Size 8 bits

10.1.17 PCI Minimum Grant

The PCI minimum grant is the minimum burst period required by the ADMA.

Address Offset 3Eh
Default Value xxh in units of 250 ns.
Attribute Read Only
Size 8 bits

10.1.18 PCI Maximum Latency

The default value of 00h indicates that the ADMA has no particular requirement for Maximum Latency.

Address Offset 3Fh
Default Value 00h
Attribute Read Only
Size 8 bits

10.1.19 Power Management Registers

Address Offset 50h
Default Value see table
Attribute see table
Size 8 Bytes

Pointer to Next Capability (Read Only)								Capability ID (Read Only)								50h
0h								01h								
Power Management Capability (Read Only)																52h
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
0	0	1	0	0	1	0	0	0	0	0	0	1	0	1	0	
Power Management Control/Status																54h
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
0	0	0	0	0	0	0	R/W	0	0	0	0	0	0	R/W	R/W	
Data Register (Read Only)								Bridge Support Extension (Read Only)								56h
0h								0h								

Table 12 – Power Management Registers

10.1.19.1 Capability ID

The capability ID indicates that the ADMA supports the {PCI PMS}.

10.1.19.2 Pointer to Next Capability

The next capability points to the next capability; 0h indicates the end of the linked list of capabilities.

10.1.19.3 Power Management Capability

Bit	Description
15:14	Reserved.
13	Set to one indicates that the ADMA may assert PME# from the D2 state if the signal UINTRQ (unsolicited interrupt) is asserted on either channel.
12:11	Reserved.
10	Set to one indicates that the ADMA supports the D2 (Standby) state.
9:4	Reserved.
3	Set to one indicates that the ADMA requires a PCI clock to assert PME#.
2:0	Set to 010b indicates that the ADMA complies with version 1.1 of the {PCI PMS}.

10.1.19.4 Power Management Control/Status

Bit	Description
15	Indicates whether PME# can be asserted from power state D3-Cold. Fixed to zero indicating that the ADMA does not support PME# assertion from the D3-Cold state.
14:9	Reserved.
8	Controls the enable and disable of PME#.
7:2	Reserved.
1:0	Power Management State Control Bits.

Power Management State Control Bits:

b1	b0	State
0	0	Active
0	1	Invalid
1	0	Standby
1	1	Sleep

Table 13 – Power Management State Control Bits

10.1.19.5 Power Management State Transitions

Bits b0 and b1 are defined in Table 13:

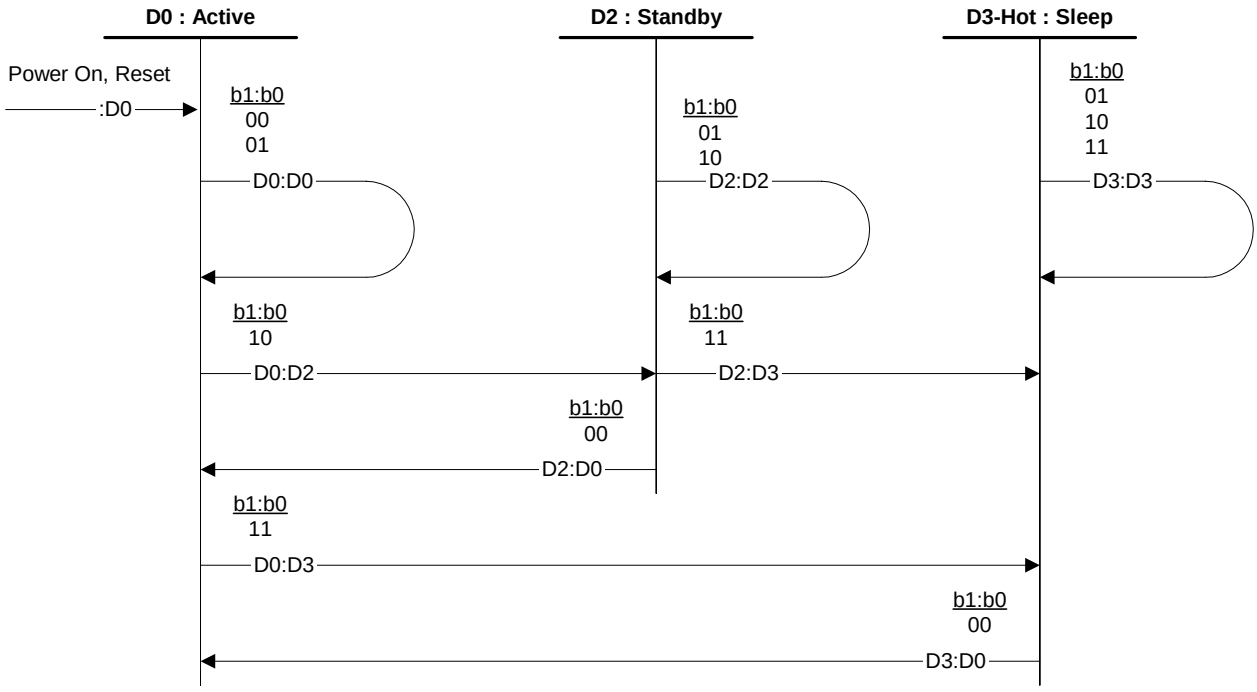

Figure 7 – Power Management State Transitions

11 ADMA Registers

The ADMA register map uses 1024 bytes of memory space. This includes the ATA Command, Control and Data port Registers. All registers may be accessed as byte, word, or Dword entities. These registers are addressed via the base address in BAR 4 (Bytes 20h-27h) in the PCI configuration header registers. The description of the ADMA registers follows.

Bits 31-24	Bits 23-16	Bits 15-8	Bits 7-0	Offset
Reserved		Channel X PIO Data		00h
Reserved			Ch. X Err/Features	04h
Reserved			Ch. X Sector Cnt	08h
Reserved			Ch. X Sector Nbr	0Ch
Reserved			Ch. X Cyl. Low	10h
Reserved			Ch. X Cyl. High	14h
Reserved			Ch. X Dev. Head	18h
Reserved			Ch. X Stat/Cmd	1Ch
Reserved				20h-37h
Reserved			Ch. X Alt Stat/Ctrl	38h
Reserved				3Ch
Reserved		Channel Y PIO Data		40h
Reserved			Ch. Y Err/Features	44h
Reserved			Ch. Y Sector Cnt	48h
Reserved			Ch. Y Sect. Nbr	4Ch
Reserved			Ch. Y Cyl. Low	50h
Reserved			Ch. Y Cyl. High	54h
Reserved			Ch. Y Dev. Head	58h
Reserved			Ch. Y Stat/Cmd	5Ch
Reserved				60h-77h
Reserved			Ch. Y Alt Stat/Ctrl	78h
Reserved				7Ch
Reserved	Ch. X ADMA Stat	Chan. X ADMA Control		80h
Reserved		Chan. X ADMA CPB Search Count		84h
Chan. X ADMA Current CPB Address				88h
Chan. X ADMA Next CPB Address				8Ch
Chan. X CPB Lookup Table Address				90h
Chan. X ADMA FIFO Output Threshold		Chan. X ADMA FIFO Input Threshold		94h
Reserved				98h-9Fh
Reserved	Ch. Y ADMA Stat	Chan. Y ADMA Control		A0h
Reserved		Chan. Y ADMA CPB Search Count		A4h
Chan. Y ADMA Current CPB Address				A8h
Chan. Y ADMA Next CPB Address				ACh
Chan. Y CPB Lookup Table Address				B0h
Chan. Y ADMA FIFO Output Threshold		Chan. Y ADMA FIFO Input Threshold		B4h
Reserved				B8h- BFh
Vendor Specific				C0h
Vendor Specific				C4h
Reserved				C8h-D3h
BIOS Message Pointer				D4h
Reserved				D8h
Driver Message Pointer				DCh
Vendor Specific				E0h-17Fh
Reserved				180h-3FFh

Table 14 – ATA and ADMA Memory Mapped Registers

11.1 ADMA Control Register (ADMCTL)

Address offset, Channel X: Base + 80h
Address offset, Channel Y: Base + A0h
Default value 100h
Attribute Read/Write
Size 16 bits

Bit	Name	Reset	Description
15:9		0	Reserved.
8	aIEN	1	PCI channel interrupt enable bit. When set to one, interrupts generated by the channel are propagated through to the PCI bus. When cleared to zero, interrupts generated by the channel are not propagated to the PCI bus
7	aGO	0	ADMA GO Bit. When set to one, the ADMA can run. When cleared to zero, the channel operates only in Legacy PIO Mode. The host writes a one to this bit each time that a CPB has been updated, to notify the ADMA that there is another CPB to service. Note: When this bit is cleared to zero by the host, the ADMA immediately ceases all operations and goes to Legacy PIO Mode; the state of the current CPB is indeterminate.
6	aPSE	0	ADMA PAUSE Bit. When set to one, the ADMA does not follow the CPB chain nor access the CPB Lookup Table. If set to one while a CPB is being processed, the ADMA completes the CPB and then PAUSES. The S/W driver shall pause operations before modifying the CPB chain pointers by the use of aPSE.
5	aRSTADM	0	RESET ADMA Channel to the IDLE state. Set to one by the host to indicate a reset is required. Cleared by the host after 1µs to allow the ADMA to come out of the IDLE state.
4			Vendor Specific.
3	aAUTEN	0	ADMA AUTO-POLL ENABLE Bit. When set to one, the ADMA repeatedly alternates selection of each device on the channel when waiting for interrupts, in order to detect the first device interrupting in an overlapped or queued situation.
2	aRSTA	0	ATA HARD RESET Bit. When set to one, the ATA reset signal is asserted. For the host to reset the ATA channel, the host shall set this bit to one, wait for the minimum reset time defined in the {ATA Spec}, and then clear this bit to zero.
1:0	aPIOMD	00	DEFAULT PIO MODE. Used in Legacy PIO Mode to define the ATA PIO timing. 00 = Mode 1, 01 = Mode 2, 10 = Mode 3, 11 = Mode 4. ATA mode zero is not supported. The value in this register shall be the highest PIO mode supported by the slowest device on the channel. The mode selected is used for all accesses to the ATA command and control block registers, in Legacy PIO and ADMA Mode. When in Legacy PIO Mode, the PIO mode selected is used for access to the ATA data port (PIO mode).

Table 15 – ADMA Control Register

11.2 ADMA Status Register (ADMSTAT)

Address offset, Channel X: Base + 82h
Address offset, Channel Y: Base + A2h
Default value See table
Size 8 bits
Attribute Read/Clear

Can be read by the host at any time. Reading the register clears bits 0, 1, and 7 to zero and de-asserts the PCI INTA signal.

Bit	Name	Reset	Description
7	aDONE	0	ADMA DONE Bit. When set to one, indicates the ADMA has finished one or more CPBs.
6	aPSD	1	ADMA PAUSED Bit. When set to one, indicates the ADMA has STOPPED as a result of aPSE being set. The current transfer has been completed.
5	aSTPD	1	ADMA STOPPED Bit. When set to one, indicates the ADMA has STOPPED as a result of aGO being cleared, an error occurring, or no more valid CPBs to be processed. If aGO has been cleared the status of the current transfer may be undefined. This bit is cleared to zero when aGO is written with a one. If a Released CPB is being processed aSTPD is cleared to zero and set to one on completion.
4	aUIRQ	X	ATA UNSOLICITED IRQ Bit. When set to one, indicates the ATA unsolicited interrupt line is active.
3	aLGCY	1	ADMA LEGACY Bit. When set to one, indicates that the ADMA is in Legacy PIO Mode.
2		0	Reserved.
1	aCPBERR	0	ADMA CPB Error Bit. When set to one, indicates that at least one of the CPB-error response flags in the CPB has been set to one except in the case of cPSEXC (Table 17) and pIGEX (Table 19) set to one.
0	aPERR	0	PCI ERROR Bit. When set to one, indicates that a PCI error has occurred.

Table 16 – ADMA Status Register

11.3 ADMA CPB Search Count Register (CCNT)

Address offset, Channel X: Base + 84h
Address offset, Channel Y: Base + A4h
Default value 0000h
Attribute Read/Write
Size 16 bits

The ADMA CPB Search Count Register holds a value that is copied into the Internal Down-Counter each time either a write occurs to ADMCTL with the ADMA aGO Bit set or the ADMA reads a CPB with the cVLD Bit set and the cDONE Bit clear. The down-counter decrements each time the ADMA reads a CPB with the cVLD Bit clear, cREL set to one or the cDONE Bit set to one. When the down counter reaches zero, the ADMA sets aDONE and transitions to the IDLE state. The value loaded into this register is normally (but not necessarily) related to the number of CPBs in the chain. This value shall be greater than zero.

11.4 ADMA Current CPB Address (CCPB)

The ADMA Current CPB Address Register points to the address of the CPB currently being processed. It is loaded by the ADMA whenever a CPB is read (or, in the case of queued operation, reads the CPB Lookup Table).

Address offset, Channel X: Base + 88h
Address offset, Channel Y: Base + A8h
Default value 00000000h
Attribute Read Only
Size 32 bits

11.5 ADMA Next CPB Address (NCPB)

Address offset, Channel X: Base + 8Ch
Address offset, Channel Y: Base + ACh
Default value 00000000h
Attribute Read/Write
Size 32 bits

The ADMA Next CPB Address Register is initialized by the host to point to the 'first' CPB in a circular chain that it has constructed in memory. The address shall be a *physical address.* This register is updated by the ADMA each time it reads a CPB, except when retrieving a CPB from the Lookup Table. The host shall not write to this register unless the ADMA is in the Legacy Idle or the Paused State (see Section 13).

Note that each CPB shall be physically contiguous and locked in memory and that all chain pointers shall be *physical addresses*.

11.6 CPB LookUp Table Address Register (CPBLAR)

Address offset, Channel X: Base + 90h
Address offset, Channel Y: Base + B0h
Default value 00000000h
Attribute Read/Write
Size 32 bits

The CPB Lookup Address Register contains the 32-bit address of the base of the CPB Lookup Table. The host software initializes the contents of this register before entering ADMA Mode. The ADMA uses this address as a base to which it adds the offset calculated from the sub-channel, device, and tag resulting from a Service command. The resulting calculated address points to a location in memory that, in turn, points to the CPB associated with the command requiring service.

The CPB Lookup Table Address Register is only used in overlapped or queued operation. It is initialized by the host to point to the base of a Lookup Table that it has constructed in memory. The CPB Lookup Table shall be physically contiguous and locked in memory. The address contained in this register shall be a *physical address*. This register is used by the ADMA to construct the address of the applicable CPB when a service interrupt is received. Each entry in the table is a Dword holding the *physical address* of the CPB.

11.7 ADMA FIFO Input Threshold (FITR)

Address offset, Channel X: Base + 94h
Address offset, Channel Y: Base + B4h
Default value 000h
Attribute Read/Write
Size 16 bits)

The ADMA FIFO Input Threshold is initialized by the host to ensure reasonable PCI burst sizes. It represents the number of Qwords in the Input FIFO to the ADMA from the PCI bus before the FIFO Input Threshold (FIT) Flag sets and a PCI burst is initiated to write the data to memory. The value loaded into this register is normally (but not necessarily) related to the value in the CacheLine Size Register (offset 0Ch) in the PCI configuration Registers. The value of FITR shall be greater than zero and less than the FIFO size.

11.8 ADMA FIFO Output Threshold (FOTR)

Address offset, Channel X: Base + 96h
Address offset, Channel Y: Base + B6h
Default value 000h
Attribute Read/Write
Size 16 bits

The ADMA FIFO Output Threshold is initialized by the host to ensure reasonable PCI burst sizes. It represents the space, in Qwords, in the Output FIFO from the ADMA to the PCI bus before the FIFO Output Threshold (FOT) Flag sets and a PCI burst is initiated to read data from memory. The value loaded into this register is normally (but not necessarily) related to the value in the Cache Line Size Register (offset 0Ch) in the PCI configuration Registers. The value of FOTR shall be greater than zero and less than the FIFO size.

11.9 BIOS Message Pointer

Address Offset	D4h
Default Value	00000000h
Attribute	Read/Write
Size	32 bits

This register may be used by the BIOS to point to its message area. The ADMA takes no action on this register other than clearing it on reset. This allows a BIOS to provide information to a device driver.

11.10 Driver Message Pointer

Address Offset	DCh
Default Value	00000000h
Attribute	Read/Write
Size	32 bits

The host device driver may use this register to point to its message area. The ADMA takes no action on this register other than clearing it on reset. This allows a device driver to provide information to a BIOS.

12 Auto DMA Mode Data Structures

In ADMA Mode the ADMA engine reads a command set held in a command chain from host memory. Command sets are held in a data structure termed a Command Parameter Block (CPB). A circular chain of CPBs is created in memory, with each CPB pointing to the next CPB. The CPB pointer shall be a physical address. The ADMA maintains a Dword register that points to the next CPB of the chain. Within each CPB, there is a pointer to the chain of Physical Region Descriptors (PRDs). The PRD is a structure that defines the memory locations where the data is to be written to or read from. The PRD pointer shall be a physical address.

12.1 Command Parameter Block

The CPB is a block of parameters and commands for the ADMA and, indirectly, for the ATA Channel. Each CPB shall all be physically contiguous, locked in memory, and Qword-aligned in physical address space.

CPB data, described in Table 17, is written by the host; only the Response Flags are modified by the ADMA.

Qword		Byte	Bits	Name	Init	Description
0	Response Flags	0	0	cDONE	1	The host shall clear this bit to zero to give the ADMA control of the CPB. The ADMA sets this bit to one to give the host control of the CPB. See 12.3.
			1	cREL	0	cREL is set to one by the ADMA if it receives an interrupt from the ATA device with the REL Bit set. See Section 12.3.
			2	cIGNRD	0	cIGNRD is set to one by the ADMA if, on the first access of the CPB, cVLD, cREL and cDONE are cleared to zero. See Section 12.3.
			3	cATERR	0	cATERR is set to one by the ADMA if ATA ERR is set to one during execution of the command.
			4	cSPNT	0	cSPNT is set to one by the ADMA if it detects a spurious interrupt on the ATA INTRQ signal during execution of a command.
			5	cPSDEF	0	cPSDEF is set to one by the ADMA if the PRD data transfer lengths are insufficient to complete the command.
			6	cPSEXC	0	cPSEXC is set to one by the ADMA if the PRD data transfer length is in excess of that required to complete the command.
			7	cCPBERR	0	cCPBERR is set to one by the ADMA if it determines that the CPB is inconsistent.
		1	7:0		0	Reserved.
	Control Flags	2	0	cVLD	0	cVLD is used in combination with cDONE and cREL to control the processing of the CPB by the ADMA engine. When cDONE is set to one, the CPB will not be processed. See Section 12.3.
			1	cQUE	0	cQUE shall be cleared to zero for non-queued or overlapped commands. Set to one for queued/overlapped commands.
			2	cDAT	0	cDAT is set to one to indicate that the PRD chain contains valid information. The address in cPRD shall be valid.
			3	cIEN	0	cIEN is cleared to zero to disable the command complete interrupt; set to one to allow the Command Complete interrupt.
			7:4		0	Reserved.
	ATA Length	3	7:0	cLEN	0	The length in Qwords of the ATA Register Field Area of the CPB (the total CPB length = 2 + cLEN Qwords).
	CPB	4-7	31:0	cNCPB	Next	Memory Address of the next CPB. Shall be Qword aligned.
1	PRD	0-3	31:0	cPRD		Memory Address of the first PRD for this CPB. Shall be Qword aligned.
		4-7			0	Reserved.
2	CMD	0-1	15:0	ATADA0	0's	ATA Register Field – see 12.1.6
	CMD	2-3	31:16	ATADA1	0's	ATA Register Field – see 12.1.6
	CMD	4-5	47:32	ATADA2	0's	ATA Register Field – see 12.1.6
	CMD	6-7	63:48	ATADA3	0's	ATA Register Field – see 12.1.6
…	CMD	…	…	…	…	…
n	CMD	0-1	15:0	ATADA0	0's	ATA Register Field – see 12.1.6
	CMD	2-3	31:16	ATADA1	0's	ATA Register Field – see 12.1.6
	CMD	4-5	47:32	ATADA2	0's	ATA Register Field – see 12.1.6
	CMD	6-7	63:48	ATADA3	0's	ATA Register Field – see 12.1.6

Table 17 – CPB Structure

12.1.1 Response Flags (Byte 0)

These flags are written by the ADMA during the processing of the CPB. Bit 0 (cDONE) shall be set to one by the host when preparing the CPB. The host shall clear the byte to zero to indicate to the ADMA that the CPB is valid and ready to be processed. The Response Flags are in a byte by themselves, so that the ADMA does not have to do a read/modify/write operation.

12.1.1.1 cDONE – ATA Command Complete Flag (Bit 0)

This flag is set to one by the ADMA hardware when it has completed processing the command in this CPB entry. It is used by the ADMA to prevent processing the CPB again on subsequent passes around the CPB chain. When set, the host has control of the CPB.

The host sets cDONE to one when initializing a CPB in the CPB chain. The host sets cVLD to one and clears cDONE to zero to indicate to the ADMA that the CPB contains valid command information. Thereafter, with the exception of cVLD, the host shall not change anything in the CPB until cDONE is set to one by the ADMA. The host can then write new command information to the CPB, set cVLD to one and, finally, clear cDONE to zero. See Section 12.3.

12.1.1.2 cREL – ATA Release Interrupt Flag (Bit 1)

This flag is set to one by the ADMA hardware when the ATA REL Bit is set to one after a queued command has been written to the device. When the ADMA has set this flag it proceeds to the next Valid-Waiting CPB, unless the ATA SERV Bit is set to one. In this latter case, the ADMA then issues a service command to the device to process a previously queued command. See Section 12.3.

12.1.1.3 cIGNRD – CPB Ignored (Bit 2)

If the ADMA hardware reads a CPB with both cDONE and cVLD cleared to zero, it sets both cDONE and cIGNRD to one, sets aDONE, and asserts the PCI INTA signal. The CPB is ignored and the next CPB in the chain is processed. cVLD is assumed to be set to one if the CPB is being accessed due to a Service Interrupt. See Section 12.3.

12.1.1.4 cATERR – ATA Command Error Flag (Bit 3)

This flag is set to one by the ADMA hardware if it sees the ATA ERR (ATAPI CHK) Bit set in the ATA Status (or Alt Status) Register during the command. When the ADMA sets this bit, it sets aCPBERR, asserts the PCI INTA signal, and transitions to Legacy PIO Mode. The host is responsible for error recovery.

12.1.1.5 cSPNT – ATA Spurious Interrupt Error Flag (Bit 4)

This flag is set to one by the ADMA hardware if the ATA INTRQ signal is asserted unexpectedly during execution of a command. When the ADMA sets this bit, it sets aCPBERR, asserts the PCI INTA signal (irrespective of the state of cIEN), and transitions to Legacy PIO Mode. The host is responsible for error recovery.

12.1.1.6 cPSDEF – PRD Deficient Length Error Flag (Bit 5)

This flag is set to one by the ADMA hardware if the total transfer length in the PRD chain is insufficient to complete the ATA transfer. In this situation the ATA device might be hung or data might be lost. When the ADMA sets this bit, it sets aCPBERR, asserts the PCI INTA signal (irrespective of the state of cIEN), and transitions to Legacy PIO Mode. The host is responsible for error recovery.

12.1.1.7 cPSEXC – PRD Excess Length Error Flag (Bit 6)

This flag is set to one by the ADMA hardware if the transfer is complete before the PRD length expires. In this case, the device will have completed the command, with or without errors. When the ADMA sets this bit, it sets aCPBERR, may assert the PCI INTA signal, and may transition to Legacy PIO Mode depending on the state of pIGEX (see Section 12.1.7). The host is responsible for error recovery.

12.1.1.8 cCPBERR – ATA Command Error Flag (Bit 7)

This flag is set to one by the ADMA hardware if it detects an inconsistency in the CPB. When the ADMA sets this bit, it sets aCPBERR, asserts the PCI INTA signal (irrespective of the state of cIEN), and transitions to Legacy PIO Mode. The host is responsible for error recovery.

12.1.2 Control Flags (Byte 2)

These flags control the detailed operation of the ADMA sequencer. They remove the need for the ADMA to recognize the ATA command set. Thus, if new commands are defined, the ADMA can still function.

12.1.2.1 cVLD – CPB Valid (Bit 0)

The host shall set cVLD to one to indicate that the CPB will be processed when cDONE is cleared to zero. The host shall not set cDONE to one (unless it is initializing the CPB chain). If the host determines that a CPB need no longer be processed, it may clear cVLD to zero. Note that this does not necessarily guarantee that the CPB will be ignored. If the CPB is accessed by the ADMA after this bit is set, the ADMA ignores the CPB and indicates such by setting cIGNRD to one. If the CPB is in the Released State, the ADMA ignores cVLD when accessing the CPB in response to an ATA service request interrupt. To check that a command was "ignored" after cVLD has been cleared, the host shall check the state of cIGNRD after cDONE has been set to one by the ADMA. See Section 12.3 for more information.

12.1.2.2 cQUE – Overlap/Queue Flag (Bit 1)

This flag is set to one to indicate that the command set contains an overlapped/queued command. This flag cleared to zero indicates that there is no overlapped/queued command. If this flag is set, the ADMA inspects the ATA SERV Bit and the ATA REL Bit on the assertion of the ATA INTRQ signal.

12.1.2.3 cDAT – PRD Valid Flag (Bit 2)

The host shall set this flag to one to indicate that cPRD is valid. The PRD chain may contain Directed Interrupt Information, ATAPI Packet data pointers, data transfer pointers, or any combination of these.

12.1.2.4 cIEN – PCI Interrupt Enable Flag (Bit 3)

The host shall clear this flag to zero to prevent the ADMA from generating the PCI INTA signal when the command is complete. The host shall set this flag to one to allow the PCI INTA signal. Clearing this flag will not prevent the PCI INTA signal from being asserted in the event of an error.

12.1.3 cLEN – ATA Length (Byte 3)

This Byte contains the number of Qwords that follow Dword 2. This enables the ADMA to correctly request the number of Dwords to fetch for any particular CPB.

12.1.4 cNCPB - Next CPB Address (Dword 1)

The host at initialization shall construct in memory a circular chain of CPBs, each of which is physically contiguous and Qword-aligned in physical address space. Each CPB shall have in this field the physical address of the next CPB. The host shall write the address of the first CPB into the ADMA Next CPB Address Register before setting the aGO Bit in ADMCTL.

If the host needs to change the chain pointers while the ADMA is running, it shall first pause the ADMA by setting the aPSE Bit in ADMCTL, and checking that the aPSD Bit in ADMSTAT has set. This prevents the ADMA from using any a pointer that might be invalid.

12.1.5 cPRD – PRD Address (Dword 2)

The host at initialization shall construct a PRD chain, as required, so that each CPB has a corresponding PRD chain with its physical starting address in this field.

12.1.6 ATADAn ATA Register Field

The ATA Register Field is a list of Qwords describing the ATA register writes involved in a command sequence. Each Qword consists of four 16-bit entries. Each entry defines an ATA register write. There may be as many of these Qwords included in a CPB as required (see Table 18).

Control			Address					Data							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	WNB	IGN	CS1-	CS0-	DA2	DA1	DA0	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	WNB	IGN	CS1-	CS0-	DA2	DA1	DA0	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0
47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
0	WNB	IGN	CS1-	CS0-	DA2	DA1	DA0	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0
63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
END	WNB	IGN	CS1-	CS0-	DA2	DA1	DA0	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0

Table 18 – ATA Register Field

Each 16-bit entry consists of eight bits that define the register content to be written, five bits that define the address of the register to be written, with three bits being used for control purposes.

The three control bits are Ignore (IGN), Wait-Not-Busy (WNB), and End (END). IGN is used to indicate to the ADMA that this entry is to be ignored, and to skip to the next entry. WNB is used to indicate to the ADMA that

it shall wait for the device to become not busy before writing the data. The END Bit (bit 63 of the last Qword) indicates that the current entry is the last one to be processed, and shall be the last entry of a set. The ADMA reads the ATA register field entries. When the ADMA detects the END Bit set, it stops reading.

The data (DD0-DD7) and address (DA0-DA2) bits are active high (Asserted = 1). Bits CS0- and CS1- are active low (Asserted = 0).

12.1.7 PRD Chain

Each PRD Chain may contain a variable number of entries (PRDs). The PRD entry shall be physically continuous, locked in memory, and Qword-aligned in physical address space. The information in the PRD entry is derived by the host, and describes the physical addresses corresponding to the logical buffer address in the original I/O request. There can be several PRDs to describe a transfer buffer because some processors fragment physical memory by the use of paging registers.

In the case of an ATAPI Packet Command, the first PRD is used to describe the packet itself.

In the case of Directed Interrupts, the PRD contains the target Memory or I/O address and the Data to be written to the address.

12.1.8 Physical Region Descriptor

Each PRD, described in Table 19, is two Qwords in size and points to a region of memory or an I/O address. The ADMA engine reads each PRD in turn, and transfers data to or from the PRD-associated memory block or I/O address, until the ATA device interrupts to indicate the end of the transfer.

Qword	Byte	Bits	Name	Description
0	3:0	31:0	pMAD	Physical address of the start of a physically contiguous memory region. Shall be Qword-aligned. If an I/O transfer, the I/O address of the source or destination of the data.
	7:4	31:0	pLEN	If pPKT is cleared to zero, pLEN indicates the length, in Qwords, of the transfer segment. If pPKT is set to one and pDINT is cleared to zero, pLEN indicates the length, in words, of the total data transfer of all the subsequent PRDs (see pPKT). If pDINT is set to one, pLEN contains a 32-bit message (see pDINT).
1	0	0		Reserved.
		1	pIGEX	Ignore Data Excess. Set to one to indicate to the ADMA that data excess occurring in this PRD is not an error. This is primarily used when reading the results from certain ATAPI packet commands that return unknown or odd lengths of data. cPSEXC will be set but no error interrupt will be generated and the ADMA continues execution.
		2	pPKT	Set to one to indicate that pMAD is a pointer to a Packet (pPKLW indicates the length of the packet). pLEN indicates the total length of the transfer found in subsequent PRDs. pDINT shall be cleared to zero when pPKT is set to one.
		3	pDINT	Set to one to indicate that a Directed Interrupt (DINT) is to be performed, if a non-error interrupt event occurs. pMAD is the memory or I/O address into which to write a 32-bit message contained within pLEN. Note that the PCI INTA signal is controlled by cIEN only (both Directed Interrupts and the PCI INTA signal may be enabled, depending on the respective states of pDINT and cIEN).
		4	pORD	Data Transfer method. Set to one for Ultra-DMA, cleared to zero for DMA- assisted PIO.
		5	pDIRO	Data Transfer Direction. Shall be set to one for output from the ADMA to the ATA device, cleared to zero for input from the ATA device to the ADMA.
		6	pIOM	Set to one for I/O transfers, cleared to zero for Memory transfer.
		7	pEND	In the last PRD of a PRD chain, pEND shall be set to one and pNXT cleared to zero.
	1	3:0	pTMOD	PIO mode or Ultra-DMA mode to use, depending on pORD. PIO mode 0 is not supported, and PIO modes are decremented by one, meaning PIO mode 1 is indicated by a zero in this field, PIO mode 2 by a 1, etc. The Ultra-DMA modes (0-5) are fully supported (mode 0 = 0…mode 5 = 5).
		6:4	pCRC	If pORD is set to one, this field shall define the burst size that the ADMA will use before terminating and sending a CRC. A value of 000b indicates that the entire block is transferred. Values 001b to 111b indicate the burst size in 512-Byte units.
		7		Reserved.
	2	7:0	pPKLW	Packet length in words if pPKT is set to one.
	3	7:0		Reserved.
	7:4	31:0	pNXT	Physical address of the next PRD. In the last PRD of a PRD chain, pNXT is cleared to zero.

Table 19 – PRD Structure

If the ATA device attempts to transfer more data than is specified by the PRD chain, the ADMA sets cPSDEF and cCPBERR to one, transitions to Legacy PIO Mode, and asserts the PCI INTA signal.

If the ATA device attempts to transfer less data than is specified by the PRD chain, when pIGEX is cleared to zero the ADMA sets cPSEXC and cCPBERR to one, transitions to Legacy PIO Mode, and asserts the PCI INTA signal.

If the device has completed the command and the last PRD space is not exhausted, the ADMA transitions to Legacy PIO Mode after first setting cPSEXC and aCPBERR to one, and then asserts the PCI INTA signal.

12.2 CPB LookUp Table

If an overlapped or queued operation is required, the host shall construct a CPB Lookup Table (see Figure 6) and write its physical address in CPBLAR prior to starting the ADMA. The table shall be physically contiguous, locked in memory and Qword-aligned in physical address space. Each entry shall be a Qword with the low-order Dword containing the physical address of the CPB, and the high-order Dword cleared to zero. Note that, if the host modifies the CPB chain while the ADMA is in the Paused State (see Section 13), this table shall be updated before restarting the ADMA.

The CPB Lookup Table is used to fetch the original CPB in order to access the PRD chain that controls the transfer of the data. In this case, when an ATA interrupt is received with the ATA SERV Bit set, the ADMA retrieves the ATA TAG field from the device and uses it to construct an address within this table. The address calculation is:

Contents of CPBLAR + (DEV * 100h) + (TAG * 08h)

Where DEV is the ATA DEV Bit, and TAG is the ATA TAG.

12.3 CPB States

The CPB can be in one of four States: Not-Valid, Valid-Waiting, Valid-Processing, or Released (see Figure 8). These States are controlled by three bits in the CPB structure: cDONE, cREL, and cVLD.

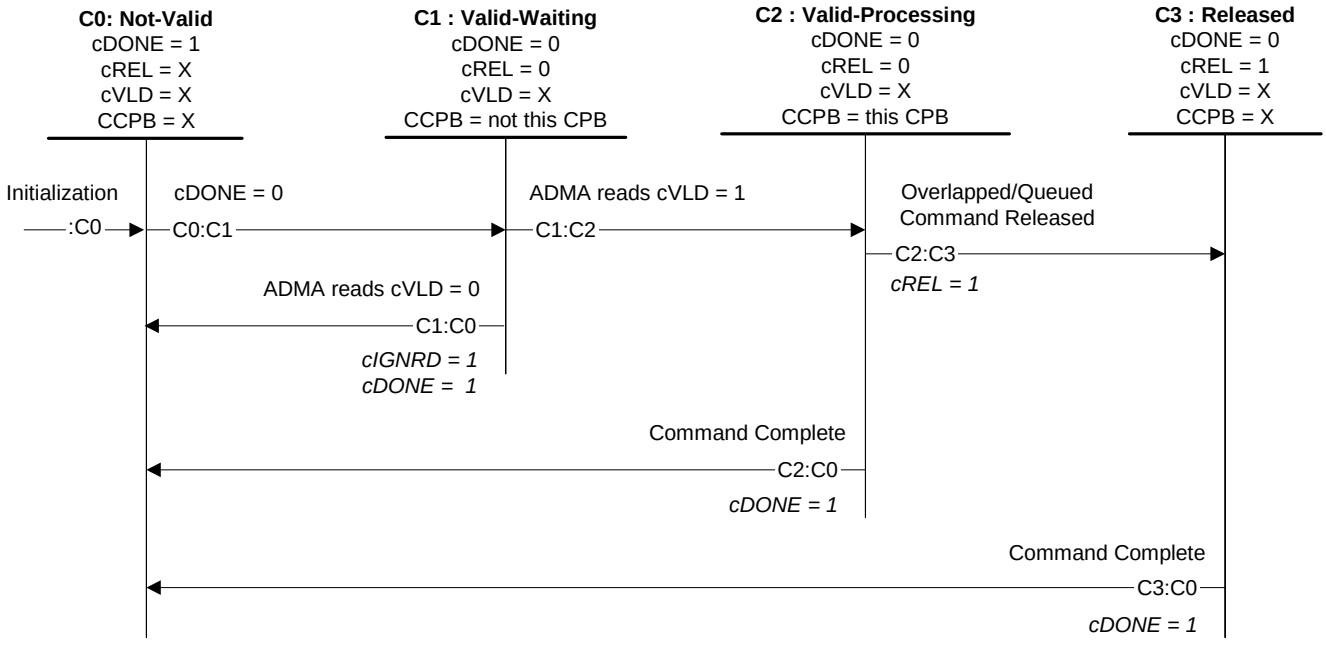

Figure 8 – CPB States

12.3.1 Not-Valid State

A CPB enters the *Not-Valid State* when cDONE is initialized to one by the host, or the ADMA sets cDONE to one at the completion of the CPB. The ADMA sets cDONE to one when the process(es) defined by a CPB have been completed, or the ADMA detects cVLD equal to zero before it begins execution of the CPB.

The host controls a CPB when it is in the Not-Valid State. The ADMA does not initiate execution of a CPB that is in this State. The host shall clear the Response Flags Byte to zero in the CPB, to transition the CPB to the Valid-Waiting State, before relinquishing control of the CPB.

12.3.2 Valid-Waiting State

A CPB enters the *Valid-Waiting State* when cDONE is cleared to zero. Normally, cVLD is set to one prior to entering this State. When the ADMA accesses a Valid-Waiting CPB, the CPB transitions to the Not-Valid State if cVLD = 0. The CPB transitions to the Valid-Processing State if cVLD = 1.

The host may cause the CPB to be ignored by clearing cVLD to zero while in the Valid-Waiting State. Note: the host is unable to differentiate between the Valid-Waiting and Valid-Processing States. This means that the host clearing cVLD to zero may or may not cause a CPB to be ignored.

12.3.3 Valid-Processing State

A CPB enters the *Valid-Processing State* when the ADMA reads cDONE = 0, cREL = 0, and cVLD = 1. In this State the ADMA delivers the ATA command(s) contained within the CPB to the ATA device. Upon completion of the ATA command(s), the ADMA sets cDONE to one, transitioning the CPB from the Valid-Processing State to the Not-Valid State. If the ATA device sets the ATA REL Bit to one, cREL is set to one, cDONE is not set, and the CPB transitions to the Released State. If the host clears cVLD to zero while in the Valid-Processing State, the ADMA engine continues processing the CPB.

12.3.4 Released State

A CPB enters the *Released State* when the ADMA sets cREL to one. In this State, cREL = 1, and cDONE = 0. The ADMA sets cREL to one when the ATA overlapped or queued command contained in a Valid-Processing CPB has been loaded into the ATA device, and the ADMA receives an interrupt from the device with the ATA REL Bit set. Upon completion of the released ATA command(s), the ADMA sets cDONE to one, transitioning the CPB from the Released State to the Not-Valid State.

13 ADMA Operation

13.1 Operational Modes and States

The ADMA engine is either in Legacy PIO or ADMA Mode. The ADMA can be in one of four States: *Legacy Idle*, *ADMA Idle*, *Run*, or *Paused*. When the ADMA is in the Legacy Idle State, the ADMA engine is in Legacy PIO Mode. When the ADMA is in any other State, the ADMA engine is in ADMA Mode.

13.2 ADMA States

The ADMA engine operates as a state machine (see Figure 9). The host software controls the ADMA state via the aGO, and aPSE Bits in ADMCTL. The host software may check the state of the ADMA via the aPSD, aSTPD, and aLGCY Bits in ADMSTAT.

13.2.1 Legacy Idle State

The Legacy Idle State is the power-on default State. In this State, the ADMA acts as an address decoder for the host. All reads and writes are performed using host I/O or host Memory instructions. The only function performed by the ADMA is to control the signal timings of the ATA bus using the ATA core, and to respond to PCI signals. In the Legacy Idle State, all data transfers use the PIO protocols, and ATA bus interrupts are directly mapped onto the PCI INTA signal.

13.2.2 Run State

In this State the ADMA reads and executes CPBs.

13.2.3 ADMA Idle State

In this State, the ADMA takes no actions.

13.2.4 Paused State

In this State, the ADMA takes no actions.

13.3 ADMA State Transitions

The ADMA is controlled by the values in ADMCTL, CCNTR, ATA ERR, and the ATA Service Interrupt. The ADMA reports its status in ADMSTAT. Figure 9 indicates the States and the transitions between them.

13.3.1 Legacy Idle to Legacy Idle (A0:A0)

The host's writing aGO as zero leaves the ADMA in the Legacy Idle State.

13.3.2 Legacy Idle to Run (A0:A2)

The host's loading a value greater than zero into CCNT and setting aGO to one will transition the ADMA from Legacy Idle to Run. CCNTR will be initialized with the contents of CCNT. The ADMA continues to execute CPBs until the Run State is exited (because aGO is cleared to zero, aPSE is set to one, or CCNTR decrements to zero).

13.3.3 Run to Run (A2:A2)

13.3.3.1 aGO is Written as One

When the host writes aGO as one, CCNTR is refreshed with the contents of CCNT. The CPB in process is not affected.

13.3.3.2 CPB Complete, aPSE = 0

When the current CPB is completed, CCNTR is decremented.

13.3.4 Run to Legacy Idle (A2:A0)

When a transition from the Run to the Legacy Idle State occurs, the host shall take the necessary steps to bring both the ATA device and the ADMA into a known state.

13.3.4.1 aGO is Written as Zero

When the host clears aGO to zero, the ADMA transitions immediately from the Run State to the Legacy Idle State. This action is not recommended: *the status of the current CPB and of the device is unknown*.

13.3.4.2 Error Condition

When an error condition occurs, the ADMA transitions from the Run State to the Legacy Idle State. This transition indicates to the host that an ADMA recognized error has occurred. See Section 13.5, Error Handling, for a complete discussion of ADMA recognized error conditions.

13.3.5 Run to Paused (A2:A3)

If aPSE and aGO are set to one when the current CPB is completed, the ADMA transitions to the Paused State.

13.3.6 Run to ADMA Idle (A2:A1)

The ADMA transitions to the ADMA Idle State when CCNTR decrements to zero. aGO is unchanged by the ADMA as a result of this transition. If a Valid-Processing CPB is completed at the same time, the ADMA sets aDONE to one, and may assert the PCI INTA signal.

13.3.7 Paused to Legacy Idle (A3:A0)

If the host clears aGO to zero, the ADMA transitions to the Legacy Idle State.

13.3.8 Paused to Run (A3:A2)

When the host sets aGO to one and clears aPSE to zero, the ADMA transitions to the Run State and continuously executes CPBs.

When the host sets aGO to one and aPSE to one, the ADMA transitions to the Run State and executes one CPB. See Section A.9.

In the event that an ATA Service Interrupt has occurred while the ADMA was in the Paused State, the ADMA may process the Service Request(s) before executing any non-Released CPB.

13.3.9 ADMA Idle to Run (A1:A2)

13.3.9.1 aGO is Written as One

When the host writes aGO as one, the ADMA engine transitions from ADMA Idle to Run, and CCNTR is initialized with the contents of CCNT. If aPSE = 1, a single CPB will be executed. If aPSE = 0, the ADMA engine executes CPBs continuously.

13.3.9.2 ATA Service Interrupt

If the ADMA receives an ATA Service interrupt the ADMA transitions from ADMA Idle to Run automatically without refreshing the value in CCNTR.

13.3.10 ADMA Idle to Legacy Idle (A1:A0)

The host's clearing aGO to zero shall transition the ADMA to the Legacy Idle State.

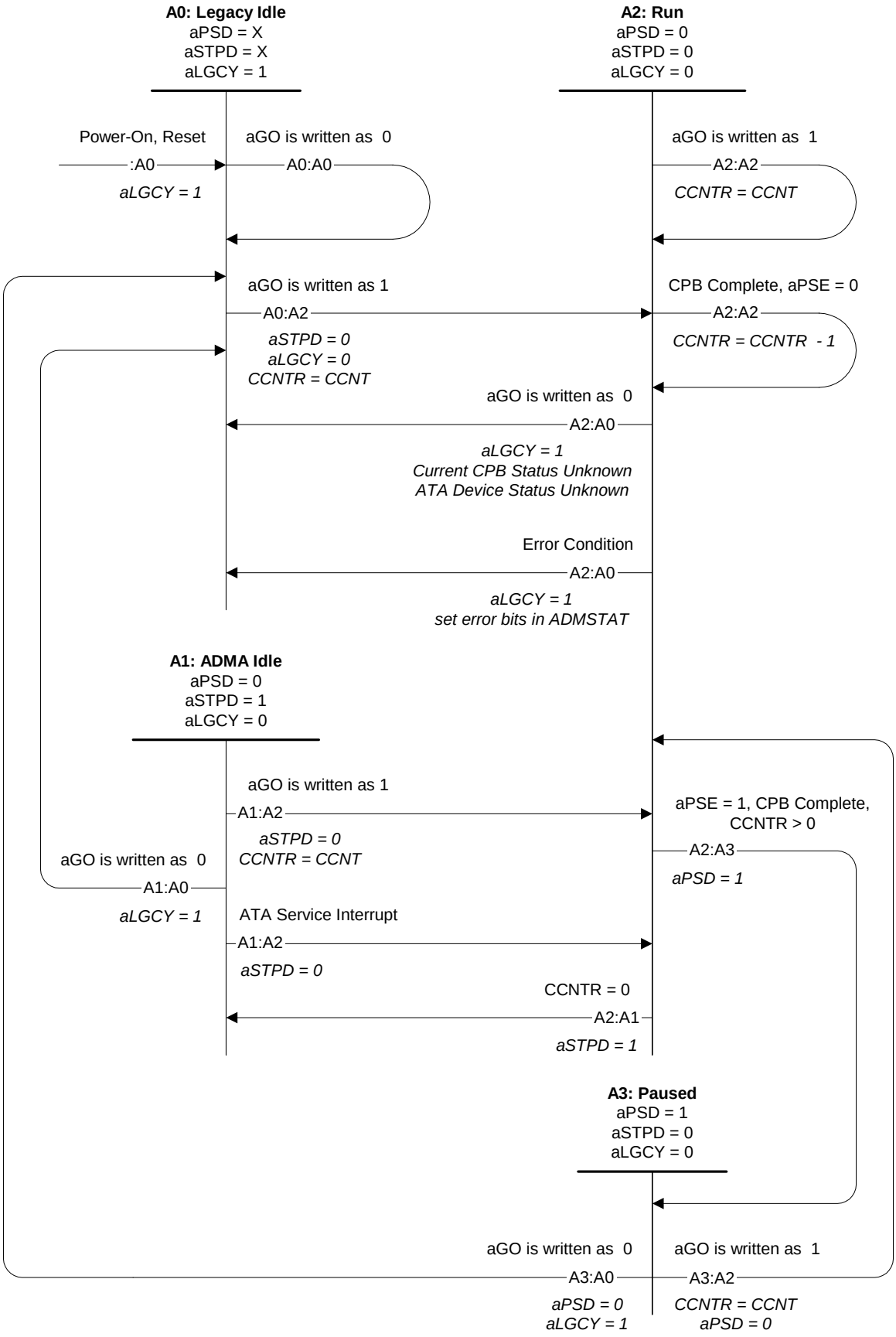

Figure 9 – ADMA State Transitions

13.4 Interrupt Assertion

The ADMA asserts the PCI INTA signal when it sets a bit in ADMSTAT. See aIEN in Table 15, and cIEN in Table 17 for exceptions. Note: if an error occurs, CCPB may not point to the CPB with the error.

13.5 Error Handling

The ADMA detects the following types of error condition: ATA Error, ATA Spurious Interrupt, CPB Error, PRD deficiency, and PRD excess. Each of these errors are reported in the CPB Response Byte.

The ADMA also detects PCI errors. A PCI Error is reported by aPERR being set to one in ADMSTAT.

In all instances of an error occurring, the ADMA transitions to the Legacy Idle State. When an error has occurred, the ADMA ignores a write of one to aGO until ADMSTAT is read, clearing aCPBERR.

13.5.1 ATA Error

When the ADMA detects ATA ERR set to one, it sets cATERR and aCPBERR to one, transitions to the Legacy Idle State, and asserts the PCI INTA Signal. (The host software shall assume that the CPB did not complete successfully.) In ATA devices, this indicates that an error has occurred. In ATAPI devices, this might indicate an error or a check condition.

13.5.2 ATA Spurious Interrupt

If the ATA INTRQ signal is unexpectedly asserted while the ADMA is in the Run State, the ADMA: sets cSPNT to one, sets aCPBERR to one, transitions to the Legacy Idle State, and asserts the PCI INTA signal.

A spurious interrupt may indicate a faulty ATA channel or a device malfunctioning. Data transfers in progress are stopped.

When an ATA spurious interrupt occurs, the host regains control of the ATA channel by toggling aRSTADM followed by aRSTA in ADMCTL.

13.5.3 CPB Error

This error occurs when a service interrupt points to a CPB that is not in the Released State. The ADMA sets cCPBERR to one, sets aCPBERR to one, transitions to the Legacy Idle State, and asserts the PCI INTA signal. This means an ATA TAG, or an ADMA data structure, has been corrupted.

13.5.4 PRD Deficiency

This error occurs when the PRD transfer lengths are insufficient to complete the command. The ADMA sets cPSDEF to one, sets aCPBERR to one, transitions to the Legacy Idle State, and asserts the PCI INTA signal.

13.5.5 PRD Excess

This error occurs when the PRD transfer lengths are in excess of that required to complete the command. The ADMA sets cPSEXC to one irrespective of the state of pIGEX. If pIGEX is cleared to zero in the PRD, the ADMA sets aCPBERR to one in ADMSTAT, transitions to the Legacy Idle State, and asserts the PCI INTA signal.

13.5.6 PCI Error

The ADMA detects a PCI error whenever bits 8, 12, 13, or 15 of the PCI Status Register are set to one, indicating a severe system problem. See {PCI Spec}. Any transfers across the PCI bus may result in catastrophic failure. The ADMA ceases all ATA operations, sets aPERR to one, transitions to the Legacy Idle State, and asserts the PCI INTA signal. The ADMA does not attempt to update the CPB, as this would involve a complete master mode operation on the suspect PCI bus. The host software shall take whatever actions it can to determine the state of the bus, before attempting any accesses to the ADMA.

The PCI INTA interrupt signal will remain asserted until aPERR is cleared to zero by a read of ADMSTAT.

Appendix A Programming Guidelines (Informative)

A.1 Introduction

This section is intended to review some aspects of programming in ADMA Mode that might not be immediately obvious. It makes the assumption that readers have familiarized themselves with the rest of the document.

In the past, Intel x86 PC Legacy ATA controllers have been accessed using specific I/O addresses (1Fxh, 17xh) and a specific set of IRQs (14, 15) where each ATA channel has a dedicated IRQ. The ADMA controller in its Legacy PIO Mode uses a *single interrupt for both channels*. Therefore, the host should not place the I/O BARs at the Legacy I/O addresses as this may cause the host OS to confuse the ADMA controller with an ATA Legacy controller. This is why the ADMA uses a PCI sub-class code of 05h as opposed to 01h (see Section 10.1.6).

A.2 Programming the ADMA

The ADMA facilitates transfer of commands and data between the PCI bus and the ATA bus. This is done by a command chaining technique discussed in Section 8.1. This command chain is dependant upon the ADMA hardware, *and data structures residing within system memory*. It is the responsibility of the host software to allocate and initialize these structures before commencing data transfers via the ADMA.

A.2.1 PCI Configuration Header Registers

The PCI Configuration Header Registers, discussed in Section 10.1, are initialized by the host OS or BIOS at system boot. The ADMA Registers, discussed in Section 11, for both channels X and Y are addressed through BAR 4 in the PCI configuration header registers. The device driver is not responsible for initializing the PCI Configuration Header Registers.

A.2.2 CPB Chain

The CPB chain, discussed in Section 12.1, is a circular linked-list of CPB structures. A CPB chain is constructed of *one or more* CPBs. The CPB chain should be allocated and initialized by the host software before entering ADMA Mode.

The ADMA CPB Next Address Register should be initialized with a pointer to a CPB in the chain.

ATA register fields are used by the ADMA to write to the ATA device's Command and Control Registers. The Wait-Not-Busy (WNB) Bit set to one instructs the ADMA to read the ATA device's Status Register, and wait until the device is not busy before writing the ATA register. WNB should be set to one on the first register write for each ATA command, as the ADMA may not wait for the device to become not busy before writing the ATA register. If there are fewer than four valid entries in an ATA register field, the IGN Bit is used to indicate entries to be ignored. Note that IGN should not be set to one if END is set to one (see Section 12.1.6).

A CPB structure may contain a variable number of ATA register writes (see Section 12.1.6). This number may be fixed, or may vary dynamically from CPB to CPB. A CPB contains up to one data transfer command. A CPB may contain multiple non-data commands preceding a data transfer command, if any.

A.2.3 PRD Chains

A PRD chain, discussed in Section 12.1.7, is a linked-list of PRD structures. A PRD chain is constructed of *one or more* PRDs. Each CPB containing an ATA data transfer command or packet command should point to a valid PRD chain. *An ATA command that does not involve the transfer of data does not require a PRD*.

PRD chains may be allocated before entering ADMA Mode, or at run time. The device driver is responsible for allocating and initializing one PRD chain for each CPB transferring data in the CPB chain before making the CPB valid.

It is very important to ensure that the method, direction, and length of the transfer as indicated in the PRDs are consistent with the ATA command within the CPB. *If this consistency is not maintained, the system may hang.*

A.2.4 CPB Lookup Table

If overlapped or queued commands are to be used, the device driver is responsible for allocating and initializing a CPB Lookup Table, discussed in Section 0. The CPB Lookup Table should be allocated and initialized before entering ADMA Mode. Each entry should be a Qword, with the low-order Dword containing the physical address a CPB, and the high-order Dword cleared to zero.

The ADMA CPB Lookup Address Register should be initialized with a pointer to the CPB Lookup Table.

This table should be updated before restarting the ADMA if the host modifies CPB the chain (see A.11).

A.3 Asynchronous Operation

The ADMA operates asynchronously from the host. This means that the ADMA may process more than one command in between the host being able to service interrupts, or even as the host is servicing one. When

servicing an interrupt, the host software should search the entire CPB chain to determine if more than one CPB has been completed (with or without errors).

The host should be aware that any updates it may make to shared system memory (see Figure), including the CPBs, PRDs, CPB Lookup Table, data buffer areas, and also the ADMA registers “CPB Next Address Register”, and “CPB Lookup Table Address Register” must be undertaken such that they will not be accessed by the ADMA at the same time (see A.11).

A.4 Memory Alignment

The ADMA is designed to use *Quad Word alignment*. Any transfer requests that the ADMA receives from the host should be aligned to a Quad Word boundary. This may mean that a device driver has to copy the data to/from an internal Quad word aligned buffer before/after the ADMA transfer.

A.5 Register Usage

The ADMA provides a set of I/O and memory mapped registers. The I/O mapped registers provide support for Legacy PIO Mode operation only, and are intended for use during the initial boot process. The memory mapped registers provide a shadow of the I/O registers, as well as the registers needed to control ADMA operation. *Device driver writers are encouraged to use the memory mapped registers, as the I/O mapped registers may be made obsolete in future versions of the ADMA*.

A.6 Legacy PIO Mode

The ADMA defaults to Legacy PIO Mode at power up, system reset, and upon detecting an error. In Legacy PIO Mode the ADMA is acting as an address decoder and ATA bus timing device only. The host reads and writes an ATA device just as it would on a legacy ISA bus adapter. The I/O mapped registers are initialized by the host OS or BIOS, and usually will not be the legacy 1Fxh/3Fxh values. The ADMA *does not* provide separate interrupts for each channel, and thus host software in Legacy PIO Mode should use the PCI shared interrupt architecture. Legacy PIO Mode is primarily used for error handling; device drivers should poll the ATA registers and not rely on interrupts.

A.7 Resets

A.7.1 PCI Reset

PCI Reset resets the ADMA PCI Core, PCI sequencer, ATA sequencer, and asserts the ATA RESET signal.

A.7.2 ADMA Reset (aRSTADM)

aRSTADM resets the ATA and PCI state machines; this is the same as power-on. Asserting aRSTADM sets the state machines to their idle state (see Section 11.2). Registers and data transfers are in an unknown state. The host should reinitialize all ADMA register values (not PCI register values) for this channel. The devices on this channel should then be reset and reinitialized.

A.7.3 ATA Channel Reset (aRSTA)

Setting aRSTA to one asserts the ATA reset signal, clearing aRSTA to zero de-asserts the ATA reset signal. The interval between the two should be at least the minimum specified in the relevant ATA standard.

A.8 Use of aGO

Writing a one to aGO can be thought of as a “door bell”. The intent is to signal to the ADMA that the host has changed something in the CPB chain, or in ADMCTL. It does not matter that the value of aGO is already one, it is the act of writing a one into aGO in ADMCTL that provides this indication to the ADMA.

A.9 Execute Single CPB

In a development environment, it is often useful to “single-step” through a CPB chain. This is achieved by setting aPSE to one and writing aGO as one for each CPB to be executed. Note: if a service interrupt is pending, the CPB that is executed may not be the next one in the chain.

A.10 Determining the Current Status of the ADMA

The host determines the status of the ADMA by examining ADMSTAT. Figure 9 shows the expected outputs from the status register dependent on the ADMA’s current state. It should be noted that when the ADMA is in the ADMA Idle State, there might be outstanding Released CPBs. This means that the state of the ADMA may change without any action by the host.

A.11 Host Pausing of the ADMA Engine

The host may pause ADMA processing of an active chain, at any time, by setting aPSE to one. Upon completion of the current CPB, the ADMA sets aDONE to one, cDONE to one, transitions to the Paused State, and may assert the PCI INTA signal. If the ADMA is releasing a CPB it sets cREL to one and transitions to the Paused State without asserting the PCI INTA signal.

The host may determine when the ADMA has transitioned into the Paused State by checking aPSD set to one.

The host may determine which CPB was just completed by reading the ADMA Current CPB Address register (CCPB). The host may then modify any of the non-Released CPB entries in the chain. The host should take care to ensure that any CPBs that are in the Released State are not invalidated by changes made to the chain or to the CPB Lookup Table.

By leaving aPSE set to one, the host may execute the next valid CPB by writing a one to aGO. See Section A.9. The host may continue completion of all valid CPBs in the chain by clearing aPSE and writing a one to aGO.

A.12 Host Stopping or Terminating an Active CPB

The host may force the ADMA into Legacy PIO Mode by clearing aGO regardless of the current state of the ADMA engine. This action immediately stops the processing of the CPB chain. *If a CPB is being processed, the results will be indeterminate, and the state of the ATA device will be unknown.* By pausing the ADMA first, the host may make an orderly transition to Legacy PIO Mode. The host may then examine the CPB chain to determine if any Released CPBs exist, and if so, take appropriate action. The host should issue an ATA channel reset to bring the device(s) on the stopped channel to a known condition.

A.13 ATA Interrupts

The ADMA relies on the use of the ATA device interrupt INTRQ. Under no circumstances should the host software set nIEN to one, thereby disabling the ATA INTRQ signal. The ADMA engine will not operate correctly in ADMA Mode with this bit set to one.

A.14 ADMA Interrupts

The host may determine if the ADMA caused an interrupt by examining ADMSTAT. An interrupt has occurred when aDONE and/or an error bit (aUIRQ, aCPBERR, aPERR) is set to one (see Section 13.4). Reading ADMSTAT clears all error bits and aDONE to zero and de-asserts the pending interrupt.

A.15 Chain Management

The host should ensure that there is a correctly initialized CPB chain, and if overlapped/queued operations are required a correctly initialized CPB Lookup Table, before entering ADMA Mode. A valid CPB chain should consist of one or more CPB structures with the Next CPB fields pointing to the physical memory address of the next CPB in the chain (the Next CPB field in a chain of one CPB would point to itself). cDONE should be set to one and cREL should be cleared to zero in each CPB. The host should write the address of the first CPB into the ADMA Next CPB Address register, and the start of the contiguous CPB Lookup Table into the ADMA Lookup Table Address register.

When a CPB is ready to be processed, the host should ensure that the cREL Bit is cleared and cVLD is set to one before clearing cDONE to zero. Once cDONE is cleared to zero the ADMA is in control of the CPB. The host should not modify any CPB with cDONE cleared to zero unless the ADMA is in the Legacy Idle or Paused State. If the ADMA is in the ADMA Idle State, the host should check any CPB before modification to ensure that *cREL has not been set to one*. Such a CPB is in the “Released” State, and should not be manipulated by the host until cDONE has been set to one by the ADMA (see 12.3).

While cDONE is equal to zero, the host may attempt to stop a CPB from being processed by setting cVLD to one. The host should then wait to ensure that cDONE has been set to one before modifying the CPB. If a CPB is being processed when cVLD is cleared to zero, the CPB will continue to be processed to completion by the ADMA, and cIGNRD will not be set (see Section 12.1.1.3).

A.16 Error Handling

If the ADMA detects that the ATA Error Bit has been set the ADMA sets the appropriate error bits in the CPB and ADMSTAT, transitions to Legacy PIO Mode, and asserts an interrupt.

If the ADMA detects a PCI error, it is an indication of a severe system problem. Any transfers across the PCI bus are now suspect and may result in catastrophic failure. The ADMA ceases all ATA operations, sets aPERR in ADMSTAT to one, transitions to Legacy PIO Mode, and asserts the PCI INTA signal. The ADMA does not attempt to update the CPB, as this would involve a complete master mode operation on the suspect PCI bus.

The host software should take whatever actions it can to determine the state of the bus before attempting any more accesses to the ADMA.

Other errors indicate some kind of CPB inconsistency. Data Insufficiency, Excess, and CPB error (bits cPSDEF, cPSEXC, cCPBERR in the CPB Response Byte, see Section 13.5) usually mean that the CPB and PRD were not correctly constructed, or that there has been some type of data transfer error. The one exception is data excess during a Packet data transfer. In some cases the transfer size is not known or is not an exact Qword in length. In such cases the PRD data transfer length should be rounded up to the nearest Qword and pIGEX in the PRD set to one. In this way, the ADMA does not stop on error (see Section 13.5.5.).

A.17 ATAPI Data Transfers

All ATAPI PRD chains associated with a data transfer should contain at least two PRDs. The first PRD points to the packet data and should have pDIRO set to one to indicate output to the ATA device, and pORD cleared to zero to indicate use of the PIO protocol. The subsequent PRDs should indicate the direction and transfer mode for the associated data (see Section 12.1.7).

A.18 Queued Operation

For channels supporting two overlapped/queued devices the Auto-Poll Enable (aAUTEN) bit should be set to one. This causes the ADMA to alternately select each device when the ATA bus has been released, so that either device that requires service has an opportunity to assert an interrupt.

The ATA Standard suggests that nIEN be toggled during the queued protocol. This should not be done. See Section A.13.

During queued operation, an ATA error will abort the internal command queue in the ATA device. All Released CPBs for that device will need to be reissued.