**Working Draft**

**T13 1532D Volume 2**

**Revision 4**
**23 Dec 2003**

# Information Technology - AT Attachment with Packet Interface - 7 Volume 2 - Parallel Transport Protocols and Physical Interconnect (ATA/ATAPI-7 V2)



T13 Technical Editor:

John Masiewicz
Western Digital
20511 Lake Forest Drive
Lake Forest, CA 92630
USA

Tel: 949-672-7686
Fax: 9496725499

Reference number
ANSI INCITS.*** - xxxx
Printed December, 24, 2003 12:13PM

Other Points of Contact:

T13 Chair
Dan Colgrove
Hitachi Global Storage Technologies
2903 Carmelo Dr
-Henderson, NV 89402
Tel: 702-614-6199
Fax: 702-614-7955

T13 Vicechair
Jim Hatfield
Seagate Technology
389 Disc Drive
Longmont CO 80503
Tel: 720-684-2120
Fax: 720-684-2711

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

T13 Reflector

See the T13 Web Site at http://www.t13.org for a description of the FTP site.

T13 Web Site

http://www.t13.org

T13 Anonymous FTP Site

ftp.t13.org

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 0 - November 5, 2001

Document created from ATA/ATAPI-6-revision 2a (T13/1410Dr2a).
Added editorial changes requested for ATA/ATAPI-6 at the October 23-25, 2001 plenary meeting except the removal of the Streaming feature set.
Formatted as discussed at the October 23-25,2001 plenary including moving connector specs to the body of volume 3 and adding form factor descriptions to volume 3.

Revision 0a- December , 2001

Revised format as requested at the December 11-12, 2001 plenary meeting.
Made changes agreed to for the resolution of letter ballot comments for ATA/ATAPI-6.

Revision 0b - February 27, 2002

Made changes requested by the ANSI editor for ATA/ATAPI-6.
Made changes resulting from public review comment resolution for ATA/ATAPI-6, e02109r0.
Added e01135r1, UDMA 133 as approved at the February 2002 plenary.
Added e01139r2, Selective self-test as approved at the February 2002 plenary.

Revision 0c - April 25, 2002

Added e02101r0 Proposal to obsolete SEEK as approved at the April 2002 plenary.
Reserved eight opcodes, four IDENTIFY DEVICE words, one SET FEATURES subcommand code pair, and eight log addresses for Serial ATA per e01145r1 as approved at the April 2002 plenary.

Revision 0d - July 8, 2002

Added the following proposals as approved at the June 2002 plenary:
e01119r0 Leakage current on the RESET signal
e01137r2 Conveyance self-test
e01141r2 Forced unit access commands
e02119r0 Editorial comments on the 1.8” 3.3v parallel form factor
e02123r0 Item 1 reorganizing register descriptions, Item 3 adding note about slew rate on RESET signal

Revision1 - August 28, 2002

Added the following proposals as approved at the August 2002 plenary:
e01138r3 Large physical sector size
e02115r2 ID data to support ISO 7779
e02127r AV IDENTIFY DEVICE word 96
Made AAM optional for devices implementing the PACKET command feature set as approved at the August 2002 plenary.
Made changes requested during the change bar review at the August 2002 plenary.

Revision 1a - October 31, 2002

Added the following proposals as approved at the October 2002 plenary:
e02127r1 Streaming transfer times for PIO and DMA modes
e02136r1 World Wide name definition for ATA devices
e02142r0 Modifications to the “Signal integrity and UDMA implementation guide” annex
e02139r1 IDEMA Japan response to AV command modification
Added definition for host
Made changes requested during the change bar review at the October 2002 plenary.

Revision 2 - February 18, 2003

Made the following changes approved at the December 2002 plenary:
Added e02121r1 AV Lite proposal.
Made changes requested during the change bar review
Modified world wide name layout in INDENTIFY DEVICE response
Modified Volumes 1 and 2 from cover page through clause 3 to add Serial ATA material.
Created volume 3 from Serial ATA Specification 1.0

Revision 2a - March 13,2003

Made changes recommended at the March 4-5 working group meeting.

Revision 3 - June 26 2003
No content changes, updated rev to correlate all volumes to revision 3

Revision 3a - June 26 2003
Changed Host PACKET DMA command state diagram to add transition from HPD3:INTRQ_wait to HPD4:Transfer _Data as approved at Jun 24-26, 2003 plenary meeting.

Revision 3b - There is no revision 3b. Version was skipped to correlate all volumes to the same revision.

Revision 3c - Changes from Aug 19-22 Plenary page-turner, accept 3a change.
Changed figures 46, 52, and 60 to indicate INTRQ bit is “V” in affected bits summary.

Revision 3d
Updated Glossary to match Volumes 1 and 3.
Incorporated e03129r1 - STATUS Register Bit Redefinition. (Redefinition of Obsolete bit).
Clause 11.1 added “Note: Serial implementations of ATA have different hardware reset timeout requirements, see Volume 3.”

Revision 3e Oct 22 2003
Reversed e03129r1 - STATUS Register Bit Redefinition. (Redefinition of Obsolete bit).
Accepted changes from prior revisions.

Revision 3f - 13 Dec 2003
Revised figure Host Packet DMA command State Diagram to correct errors in incorporation of change on HPD3 transition.

Revision 3g - 16-18 Dec 2003
Updated glossary to remove obsoleted terms for presently used terms Device 0 and Device 1.
Changed Dword to DWORD all places
Revised Volume 1 title to include more description.
Revised Glossary definition of Read Command and Write Command

Revision 4 - 23 Dec 2003
Editorial (non-content) changes, document clean-up for final draft.
Accepted all changes

**ANSI®**
**INCITS ***-xxxx**

American National Standard
for Information Technology —

# AT Attachment with Packet Interface - 7
# Volume 2
# **Parallel Transport Protocols and Physical Interconnect**
# (ATA/ATAPI-7 V2)

Secretariat
**Information Technology Industry Council**

Approved mm dd yy

**American National Standards Institute, Inc.**

## Abstract

This standard specifies the AT Attachment Interface between host systems and storage devices. It provides a common attachment interface for systems manufacturers, system integrators, software suppliers, and suppliers of intelligent storage devices. It includes the Packet Command feature set implemented by devices commonly known as ATAPI devices.

This standard maintains a high degree of compatibility with the AT Attachment Interface with Packet Interface - 6 (ATA/ATAPI-6), INCITS 261-2002, and while providing additional functions, is not intended to require changes to presently installed devices or existing software.

# 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 Page

## Tables Page

## Figures Page

## Annexes Page

## Foreword

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

This AT Attachment with Packet Interface - 7 (ATA/ATAPI-7) standard is designed to maintain a high degree of compatibility with the AT Attachment with Packet Interface - 6 (ATA/ATAPI-6) standard. This standard is published in three volumes.

This standard was developed by the ATA ad hoc working group of Accredited Standards Committee INCITS during 2001 and 200x. The standards approval process started in 200x. 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 INCITS 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, INCITS. Committee approval of the standard does not necessarily imply that all committee members voted for approval. At the time it approved this standard, the INCITS Committee had the following members:

, Chair
, Vice-Chair
, Secretary

***ORGANIZATION REPRESENTED*** ..... ***NAME OF REPRESENTATIVE***

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

Pete McLean, Chairman

Dan Colegrove, Vice-Chairman

Mark Overby, Secretary

Technical Committee T13 on ATA Interfaces, that developed this standard, had the following additional participants:

## Introduction

## Introduction

This standard encompasses the following:

Volume 1

Clause 1 describes the scope.

Clause 2 provides normative references for the entire standard.

Clause 3 provides definitions, abbreviations, and conventions used within the entire standard.

Clause 4 describes the general operating requirements of the command layer.

Clause 5 describes the I/O registers.

Clause 6 contains descriptions of the commands.

Clauses 7 through 12 point to the material in Volume 2.

Clauses 13 through 19 point to material in Volume 3.

Volume 2

Clause 1 describes the scope.

Clause 2 provides normative references for the entire standard.

Clause 3 provides definitions, abbreviations, and conventions used within the entire standard.

Clauses 4, 5, and 6 point to the material in Volume 1.

Clause 7 contains the electrical and mechanical characteristics.

Clause 8 contains the signal descriptions.

Clause 9 describes the general operating requirements of the physical, data link, and transport layers.

Clause 10 contains describes register addressing.

Clause 11 contains the transport protocols.

Clause 12 contains the interface timing diagrams.

Clauses 13 through 19 point to material in Volume 3.

Volume 3

Clause 1 describes the scope.

Clause 2 provides normative references for the entire standard.

Clause 3 provides definitions, abbreviations, and conventions used within the entire standard.

Clauses 4, 5, and 6 point to the material in Volume 1.

Clauses 7 through 12 point to the material in Volume 2.

Clause 13 contains a general overview of the serial interface.

Clause 14 describes the serial physical layer.

Clause 15 describes the serial link layer.

Clause 16 describes the serial transport layer.

Clause 17 describes the device command layer protocol for the serial interface.

Clause 18 describes the host command layer protocol for the serial interface.

Clause 19 describes the serial interface host adapter register interface.

Clause 20 describes the serial interface error handling

Page left intentionally blank

**AMERICAN NATIONAL STANDARD** **INCITS ***-nnnn**

American National Standard
for Information Systems —

# Information Technology —
# AT Attachment with Packet Interface - 7 – Volume 2 —
# (ATA/ATAPI-7 V2)

## 1 Scope

This standard specifies the AT Attachment Interface between host systems and storage devices. It provides a common attachment interface for systems manufacturers, system integrators, software suppliers, and suppliers of intelligent storage devices.

The application environment for the AT Attachment Interface is any host system that has storage devices contained within the processor enclosure.

Volume 1 defines the register delivered commands used by devices implementing the standard. Volume 2 defines the connectors and cables for physical interconnection between host and storage device, the electrical and logical characteristics of the interconnecting signals, and the protocols for the transporting commands, data, and status over the interface for the parallel interface. Volume 3 defines the connectors and cables for physical interconnection between host and storage device, the electrical and logical characteristics of the interconnecting signals, and the protocols for the transporting commands, data, and status over the interface for the serial interface. **Figure 1** shows the relationship of these documents. For devices implementing the PACKET command feature set, additional command layer standards are listed in **Table 1** and described in clause 2.

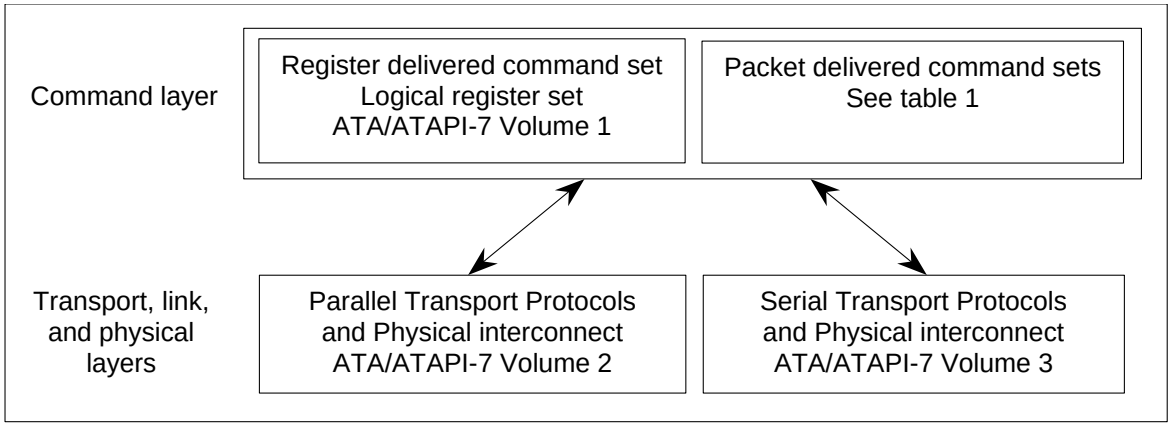



**Figure 1 – ATA document relationships**

**Table 1 – PACKET delivered command sets**

| Standard |
|---|
| SCSI Primary Commands (SPC) |
| SCSI Primary Commands - 2 (SPC-2) |
| SCSI Primary Commands - 3 (SPC-3) |
| SCSI Block Commands (SBC) |
| SCSI Stream Commands (SSC) |
| Multimedia Commands (MMC) |
| Multimedia Commands - 2 (MMC-2) |
| Multimedia Commands - 3 (MMC-3) |
| Multimedia Commands - 4 (MMC-4) |
| ATAPI for Removable Media (SFF8070I) |
| ATA Packet Interface (ATAPI) for Streaming Tape QIC-157 revision D |

This standard maintains compatibility with the AT Attachment with Packet Interface - 6 standard (ATA/ATAPI-6), INCITS nnn-200x, and while providing additional functions, is not intended to require changes to presently installed devices or existing software.

## 2 Normative references

The following standards contain provisions that, through reference in the text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards listed below.

Copies of the following documents can be obtained from ANSI: Approved ANSI standards, approved and draft international and regional standards (ISO, IEC, CEN/CENELEC, ITUT), and approved and draft foreign standards (including BSI, JIS, and DIN). For further information, contact ANSI Customer Service Department at 212-642-4900 (phone), 212-302-1286 (fax), or via the World Wide Web at http://www.ansi.org.

Additional availability contact information is provided below as needed.

### 2.1 Approved references

The following approved ANSI standards, approved international and regional standards (ISO, IEC, CEN/CENELEC, ITUT), may be obtained from the international and regional organizations who control them.

SCSI-3 Block Commands (SBC) [ANSI NCITS 306-1998] (PACKET command feature set commands)
SCSI-3 Primary Commands (SPC) [ANSI X3.301-1997] (PACKET command feature set device types)
SCSI-3 Streaming Commands (SSC) [ANSI NCITS 335-2000] (PACKET command feature set commands)
Multimedia Commands (MMC) [ANSI X3.304-1997] (PACKET command feature set sense codes)
Multimedia Commands - 2 (MMC-2) [ANSI INCITS 333-2000] (PACKET command feature set commands)
Multimedia Commands - 3 (MMC-3) [ANSI INCITS 360-2002] (PACKET command feature set commands)
Protected Area Run Time Interface Extensions (PARTIES) [ANSI INCITS 346-2001]
SCSI Primary Commands - 2 (SPC-2) [ANSI INCITS 351-2001] (PACKET command feature set commands)
AT Attachment with Packet Interface Extension (ATA/ATAPI-4), [ANSI INCITS.317-1998]
Control and Status Register (CSR) Architecture for microprocessor buses [ISO/IEC 13213:1994] (World wide names)

To obtain copies of these documents, contact Global Engineering or INCITS. Additional information may be available at http://www.t10.org and http://www.t13.org.

### 2.2 References under development

At the time of publication, the following referenced standards were still under development. For information on the current status of the document, or regarding availability, contact the relevant standards body or other organization as indicated.

SCSI Primary Commands - 3 (SPC-3) [T10/1416-D] (PACKET command feature set commands)
ATAPI for Rewritable Media [SFF8070i] (PACKET command feature set commands)
Multimedia Commands - 4 (MMC-4) [T10/1545D] (PACKET command feature set commands)

For more information on the current status of the T10 documents, contact INCITS. To obtain copies of T10 or SFF documents, contact Global Engineering.

## 2.3 Other references

The following standard and specifications are also referenced.

PC Card Standard, February 1995, PCMCIA (68-pin Connector)

For the PC Card Standard published by the Personal Computer Memory Card International Association, contact PCMCIA at 408-433-2273 or http://www.pc-card.org.

CompactFlash™ Association Specification, Revision 1.4

For the CompactFlash™ Association Specification published by the CompactFlash™ Association, contact the CompactFlash™ Association at http://www.compactflash.org.

ATA Packet Interface (ATAPI) for Streaming Tape QIC-157 revision D

For QIC specifications published by Quarter-Inch Cartridge Drive Standards, Inc., contact them at 805 963-3853 or http://www.qic.org.

EIA-364-09 TP-09C - Durability test procedure for electrical connectors and contacts
EIA-364-13 Mating and unmating forces test procedures for electrical connectors
EIA-364-17 TP-17B - Temperature life with or without electrical load test procedure for electrical connectors and sockets
EIA-364-18 Visual and dimensional inspection for electrical connectors
EIA-364-20 TP-20B - Withstanding voltage test procedure for electrical connectors, sockets, and coaxial contacts
EIA-364-21 Insulation resistance test procedure for electrical connectors, sockets, and coaxial contacts
EIA-364-23 Low level contact resistance test procedure for electrical connectors and sockets
EIA-364-27 Mechanical pulse (specified pulse) for electrical connectors
EIA-364-28 TP-28D Vibration test procedure for electrical connectors and sockets
EIA-364-31 Humidity test procedure for electrical connectors and sockets
EIA-364-32 Thermal shock (temperature cycling) test procedure for electrical conectors and sockets
EIA-364-38 TP-38B - Cable pull-out test procedure for electrical connectors
EIA-364-41 TP-41C - Cable flexing test procedure for electrical connectors
EIA-364-65 TP-65A Mixed flowing gas

For EIA specifications, contact them at www.eia.org.

# 3 Definitions, abbreviations, and conventions

## 3.1 Definitions and abbreviations

For the purposes of this standard, the following definitions apply:

**3.1.1** **ASCII Character:** Designates 8-bit value that is encoded using the ASCII Character set.

**3.1.2** **acoustics:** Measurement of airborne noise emitted by information technology and telecommunications equipment [ISO 7779:1999(E)]

**3.1.3** **ATA (AT Attachment):** ATA defines the physical, electrical, transport, and command protocols for the internal attachment of storage devices to host systems.

**3.1.4** **ATA-1 device:** A device that complied with ANSI X3.221-1994, the AT Attachment Interface for Disk Drives. ANSI X3.221-1994 has been withdrawn.

**3.1.5** **ATA-2 device:** A device that complied with ANSI X3.279-1996, the AT Attachment Interface with Extensions. ANSI X3.279-1996 has been withdrawn.

**3.1.6** **ATA-3 device:** A device that complies with ANSI X3.298-1997, the AT Attachment-3 Interface. ANSI X3.298-1997 has been withdrawn.

**3.1.7** **ATA/ATAPI-4 device:** A device that complies with ANSI INCITS 317-1998, AT Attachment Interface with Packet Interface Extensions-4.

**3.1.8** **ATA/ATAPI-5 device:** A device that complies with ANSI INCITS 340-2000, the AT Attachment with Packet Interface-5.

**3.1.9** **ATA/ATAPI-6 device:** A device that complies with ANSI INCITS 361-2002, the AT Attachment with Packet Interface-6.

**3.1.10** **ATA/ATAPI-7 device:** A device that complies with this standard.

**3.1.11** **ATAPI (AT Attachment Packet Interface) device:** A device implementing the Packet Command feature set.

**3.1.12** **AU (Allocation Unit):** The allocation unit is the minimum number of logically contiguous sectors on the media as used in the Streaming feature set. An Allocation Unit may be accessed with one or more requests.

**3.1.13** **AV (Audio-Video):** Audio-Video applications use data that is related to video images and/or audio. The distinguishing characteristic of this type of data is that accuracy is of lower priority than timely transfer of the data.

**3.1.14** **backchannel:** When transmitting a FIS, the backchannel is the receive channel.

**3.1.15** **BER (bit error rate):** The statistical probability of a transmitted encoded bit being erroneously received in a communication system.

**3.1.16** **bus release:** For devices implementing overlap, the term bus release is the act of clearing both DRQ and BSY to zero before the action requested by the command is completed. This allows the host to select the other device or deliver another queued command.

**3.1.17** **byte count:** The value placed in the Byte Count register by the device to indicate the number of bytes to be transferred during this DRQ assertion when executing a PACKET PIO data transfer command.

**3.1.18** **byte count limit:** The value placed in the Byte Count register by the host as input to a PACKET PIO data transfer command to specify the maximum byte count that may be transferred during a single DRQ assertion.

**3.1.19** **CFA (CompactFlash™ Association):** The CompactFlash™ Association that created the specification for compact flash memory that uses the ATA interface.

**3.1.20** **check condition:** For devices implementing the PACKET Command feature set, this indicates an error or exception condition has occurred.

**3.1.21 CHS (cylinder-head-sector):** This term defines an obsolete method of addressing the data on the device by cylinder number, head number, and sector number.

**3.1.22 code violation:** In a serial interface implementation, a code violation is an error that occurs in the decoding of an encoded character (see Volume 3, Clause 15)

**3.1.23 command aborted:** Command completion with ABRT set to one in the Error register and ERR set to one in the Status register.

**3.1.24 command acceptance:** A command is considered accepted whenever the currently selected device has the BSY bit cleared to zero in the Status register and the host writes to the Command register. An exception exists for the DEVICE RESET command (see clause 6) In a serial implementation, command acceptance is a positive acknowledgment of a host to device register FIS.

**3.1.25 Command Block registers:** Interface registers used for delivering commands to the device or posting status from the device. In a serial implementation, the command block registers are FIS payload fields.

**3.1.26 command completion:** Command completion is the completion by the device of the action requested by the command or the termination of the command with an error, the placing of the appropriate error bits in the Error register, the placing of the appropriate status bits in the Status register, the clearing of both BSY and DRQ to zero, and the asserting of INTRQ if nIEN is cleared to zero and the command protocol specifies that INTRQ be asserted.

**3.1.27 command packet:** A command packet is a data structure transmitted to the device during the execution of a PACKET command that includes the command and command parameters.

**3.1.28 command released:** When a device supports overlap or queuing, a command is considered released when a bus release occurs before command completion.

**3.1.29 Control Block registers:** In a parallel implementation, interface registers used for device control and to post alternate status. In a serial interface implementation, the logical field of a FIS corresponding to the Device Register bits of a parallel implementation.

**3.1.30 control character:** In a serial interface implementation, a control character is an encoded character that represents a non-data byte (see Volume 3, clause 15)

**3.1.31 CRC (Cyclical Redundancy Check):** Cyclical Redundancy Check used to check the validity of certain data transfers.

**3.1.32 Cylinder High register:** Cylinder High is the name used for the LBA High register in previous ATA/ATAPI standards.

**3.1.33 Cylinder Low register:** Cylinder Low is the name used for the LBA Mid register in previous ATA/ATAPI standards.

**3.1.34 data character:** In a serial interface implementation, a data character is an encoded character that represents a data byte (see volume 3 clause 15)

**3.1.35 data-in:** Data-in is the term used to describe the protocol that moves data from the device to the host. Such transfers are initiated by READ commands.

**3.1.36 data-out:** Data-out is the term used to describe the protocol that moves data from the host to the device. Such transfers are initiated by WRITE commands.

**3.1.37 Delayed LBA:** A delayed LBA is any sector for which the performance specified by the Streaming Performance Parameters log is not valid.

**3.1.38 device:** Device is a storage peripheral. Traditionally, a device on the interface has been a hard disk drive, but any form of storage device may be placed on the interface provided the device adheres to this standard.

**3.1.39 device selection:** In a parallel implementation,a device is selected when the DEV bit of the Device register is equal to the device number assigned to the device by means of a Device 0/Device 1 jumper or switch, or use of the CSEL signal. In a serial implementation the device ignores the DEV bit, the host adapter may use this bit to emulate device selection.

**3.1.40 disparity:** Disparity is the difference between the number of ones and the number of zeros in an encoded character (see Volume 3, Clause 15)

**3.1.41 DMA (direct memory access) data transfer:** A means of data transfer between device and host memory without host processor intervention.

**3.1.42 don't care:** A term to indicate that a value is irrelevant for the particular function described.

**3.1.43 driver:** The active circuit inside a device or host that sources or sinks current to assert or negate a signal on the bus.

**3.1.44 DRQ data block:** This term describes a unit of data words transferred during a single assertion of DRQ when using PIO data transfer.

**3.1.45 elasticity buffer:** In a serial interface implementation, the elasticity buffer is a portion of the receiver where character slipping and/or character alignment is performed.

**3.1.46 encoded character:** In a serial interface implementation, an encoded character is the output of the 8b/10b encoder (see Volume 3, clause 15)

**3.1.47 First party DMA access:** First party DMA access is a method by which a device accesses host memory. First party DMA differs from DMA in that the device sends a DMA Setup FIS to select host memory regions; whereas for DMA the host configures the DMA controller.

**3.1.48 FIS (Frame Information Structure):** The FIS is a data structure and is the payload of a frame and does not include the SOF primitive, CRC, and EOF primitive.

**3.1.49 frame:** A frame is a unit of information exchanged between the host adapter and a device. A frame consists of an SOF primitive, a Frame Information Structure, a CRC calculated over the contents of the FIS, and an EOF primitive.

**3.1.50 FUA (Forced Unit Access):** Forced Unit Access requires that user data be transferred to or from the device media before command completion even if caching is enabled.

**3.1.51 Gen1 DWORD Time:** The time it takes to transmit a 40 bit encoded value at 1.5 Gb/Sec.

**3.1.52 host:** The term host is used to describe the computer system executing the software BIOS and/or operating system device driver controlling the device and the adapter hardware for the ATA interface to the device.

**3.1.53 host adapter:** The term host adapter is used to describe the implementation of the host transport, link, and physical layers.

**3.1.54 interrupt pending:** In a parallel implementation, Interrupt pending is an internal state of a device. In this state, the device asserts INTRQ if nIEN is cleared to zero and the device is selected (see Volume 1. In a serial implementation, the interrupt pending state is an internal state of the host adapter. This state is entered by reception of a FIS with the I field set to one (see Volume 3, Clause 16)

**3.1.55 LBA (logical block address):** This term defines the addressing of data on the device by the linear mapping of sectors.

**3.1.56 LFSR (Linear Feedback Shift Register):** See volume 3 Clause 15)

**3.1.57 link:** The link layer manages the phy layer to achieve the delivery and reception of frames. See Volume 3, clause 15)

**3.1.58 logical sector:** A uniquely addressable set of 256 words (512 bytes).

**3.1.59 native max address:** The highest address a device accepts in the factory default condition, that is, the highest address that is accepted by the SET MAX ADDRESS command.

**3.1.60 overlap:** Overlap is a protocol that allows devices that require extended command time to perform a bus release so that commands may be executed by the other device (if present) on the bus.

**3.1.61 packet delivered command:** A command that is delivered to the device using the PACKET command via a command packet that contains the command and the command parameters. See also register delivered command.

**3.1.62 phy:** Physical layer electronics, See Volume 3, clause 14

**3.1.63 physical sector:** A group of contiguous logical sectors that are read from or written to the device media in a single operation.

**3.1.64 PIO (programmed input/output) data transfer:** PIO data transfers are performed by the host processor utilizing accesses to the Data register.

**3.1.65** **primitive:** In a serial interface implementation, a primitive is a single DWORD of information that consists of a control character in byte 0 followed by three additional data characters in byte 1 through 3.

**3.1.66** **queued:** Command queuing allows the host to issue concurrent commands to the same device. Only commands included in the Overlapped feature set may be queued. In this standard, the queue contains all commands for which command acceptance has occurred but command completion has not occurred.

**3.1.67** **read command:** A command that causes the device to transfer data from the device to the host (e.g., READ SECTOR(S), READ DMA, etc.).

**3.1.68** **register:** A register may be a physical hardware register or a logical field.

**3.1.69** **register delivered command:** A command that is delivered to the device by placing the command and all of the parameters for the command in the device Command Block registers. See also packet delivered command.

**3.1.70** **register transfers:** Register transfers refer to the host reading and writing any device register except the Data register. Register transfers are 8 bits wide.

**3.1.71** **released:** In a parallel interface implementation, indicates that a signal is not being driven. For drivers capable of assuming a high-impedance state, this means that the driver is in the high impedance state. For open-collector drivers, the driver is not asserted.

**3.1.72** **sector:** A uniquely addressable set of 256 words (512 bytes).

**3.1.73** **Sector Number register:** Sector Number is the name used for the LBA Low register in previous ATA/ATAPI standards.

**3.1.74** **Shadow Command Block:** In a serial interface implementation, the Shadow Command Block is a set of virtual fields in the host adapter that map the Command Block registers defined at the command layer to the fields within the FIS content.

**3.1.75** **Shadow Control Block:** In a serial interface implementation, the Shadow Control Block is a set of virtual fields in the host adapter that map the Control Block registers defined at the command layer to the fields within the FIS content.

**3.1.76** **signature:** A unique set of values placed in the Command Block registers by the device to allow the host to distinguish devices implementing the PACKET Command feature set from those devices not implementing the PACKET Command feature set.

**3.1.77** **SMART:** Self-Monitoring, Analysis, and Reporting Technology for prediction of device degradation and/or faults. Throughout this document this is noted as SMART.

**3.1.78** **transport:** The transport layer manages the lower layers (link and phy) as well as constructing and parsing FIS's. See Volume 3, clause 13

**3.1.79** **Ultra DMA burst**: An Ultra DMA burst is defined as the period from an assertion of DMACK- to the subsequent negation of DMACK- when an Ultra DMA transfer mode has been enabled by the host.

**3.1.80** **unaligned write:** An unaligned write is a write command that does not start at the first logical sector of a physical sector or does not end at the last logical sector of a physical sector.

**3.1.81** **unit attention condition:** A state that a device implementing the PACKET Command feature set maintains while the device has asynchronous status information to report to the host.

**3.1.82** **unrecoverable error:** An unrecoverable error has occurred when the device sets either the ERR bit or the DF bit to one in the Status register at command completion.

**3.1.83** **VS (vendor specific):** This term is used to describe bits, bytes, fields, and code values that are reserved for vendor specific purposes. These bits, bytes, fields, and code values are not described in this standard, and may vary among vendors. This term is also applied to levels of functionality whose definition is left to the vendor.
NOTE – Industry practice could result in conversion of a Vendor Specific bit, byte, field, or code value into a defined standard value in a future standard.

**3.1.84** **write command:** A command that causes the device to transfer data to the host to the device (e.g., WRITE SECTOR(S), WRITE DMA, etc.).

**3.1.85 WWN (world wide name):** This is a 64-bit worldwide unique name based upon a company's IEEE identifier. (See Volume 1 Clause 6 IDENTIFY DEVICE) Words (108:111).

## 3.2 Conventions

Lowercase is used for words having the normal English meaning. Certain words and terms used in this standard have a specific meaning beyond the normal English meaning. These words and terms are defined either in clause 3 or in the text where they first appear.

The names of abbreviations, commands, fields, and acronyms used as signal names are in all uppercase (e.g., IDENTIFY DEVICE). Fields containing only one bit are usually referred to as the "name" bit instead of the "name" field. (see 3.2.6 for the naming convention used for naming bits.)

Names of device registers begin with a capital letter (e.g., LBA Mid register).

The expression “word n” or “bit n” shall be interpreted as indicating the content of word n or bit n.

### 3.2.1 Precedence

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

### 3.2.2 Lists

Ordered lists, those lists describing a sequence, are of the form:

a)
b)
c)

Unordered list are of the form:

1)
2)
3)

### 3.2.3 Keywords

Several keywords are used to differentiate between different levels of requirements and optionality.

**3.2.3.1 expected:** A keyword used to describe the behavior of the hardware or software in the design models assumed by this standard. Other hardware and software design models may also be implemented.

**3.2.3.2 mandatory:** A keyword indicating items to be implemented as defined by this standard.

**3.2.3.3 may:** A keyword that indicates flexibility of choice with no implied preference.

**3.2.3.4 obsolete:**A keyword indicating that the designated bits, bytes, words, fields, and code values that may have been defined in previous standards are not defined in this standard and shall not be reclaimed for other uses in future standards. However, some degree of functionality may be required for items designated as “obsolete” to provide for backward compatibility.

Obsolete commands should not be used by the host. Commands defined as obsolete may be command aborted by devices conforming to this standard. However, if a device does not command abort an obsolete command, the minimum that is required by the device in response to the command is command completion.

**3.2.3.5 optional:** A keyword that describes features that are not required by this standard. However, if any optional feature defined by the standard is implemented, the feature shall be implemented in the way defined by the standard.

**3.2.3.6 prohibited:** A keyword indicating that an item shall not be implemented by an implementation.

**3.2.3.7 reserved:** A keyword indicating reserved bits, bytes, words, fields, and code values that are set aside for future standardization. Their use and interpretation may be specified by future extensions to this or other standards. A reserved bit, byte, word, or field shall be cleared to zero, or in accordance with a future extension to this standard. The recipient shall not check reserved bits, bytes, words, or fields. Receipt of reserved code values in defined fields shall be treated as a command parameter error and reported by returning command aborted.

**3.2.3.8 retired:** A keyword indicating that the designated bits, bytes, words, fields, and code values that had been defined in previous standards are not defined in this standard and may be reclaimed for other uses in future standards. If retired bits, bytes, words, fields, or code values are used before they are reclaimed, they shall have the meaning or functionality as described in previous standards.

**3.2.3.9 shall:** A keyword indicating a mandatory requirement. Designers are required to implement all such mandatory requirements to ensure interoperability with other products that conform to this standard.

**3.2.3.10 should:** A keyword indicating flexibility of choice with a strongly preferred alternative. Equivalent to the phrase “it is recommended”.

### 3.2.4 Numbering

Numbers that are not immediately followed by a lowercase "b" or "h" are decimal values. Numbers that are immediately followed by a lowercase "b" (e.g., 01b) are binary values. Numbers that are immediately followed by a lowercase "h" (e.g., 3Ah) are hexadecimal values.

### 3.2.5 Signal conventions

Signal names are shown in all uppercase letters.

All signals are either high active or low active signals. A dash character ( - ) at the end of a signal name indicates the signal is a low active signal. A low active signal is true when the signal is below $V_{iL}$, and is false when the signal is above $V_{iH}$. No dash at the end of a signal name indicates the signal is a high active signal. A high active signal is true when the signal is above $V_{iH}$, and is false when the signal is below $V_{iL}$.

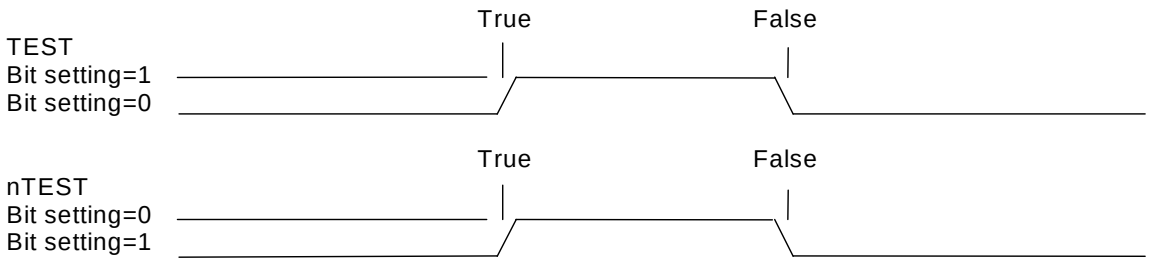
Asserted means that the signal is driven by an active circuit to the true state. Negated means that the signal is driven by an active circuit to the false state. Released means that the signal is not actively driven to any state (see clause 7). Some signals have bias circuitry that pull the signal to either a true state or false state when no signal driver is actively asserting or negating the signal.

Control signals that may be used for more than one mutually exclusive functions are identified with their function names separated by a colon (e.g., DIOW-:STOP).

SIGNAL(n:m) denotes a set of signals, for example, DD(15:0).

### 3.2.6 Bit conventions

Bit names are shown in all uppercase letters except where a lowercase n precedes a bit name. If there is no preceding n, then when BIT is set to one the meaning of the bit is true, and when BIT is cleared to zero the meaning of the bit is false. If there is a preceding n, then when nBIT is cleared to zero the meaning of the bit is true and when nBIT is set to one the meaning of the bit is false.




Bit (n:m) denotes a set of bits, for example, bits (7:0).

### 3.2.7 State diagram conventions

State diagrams shall be as shown in **Figure 2**

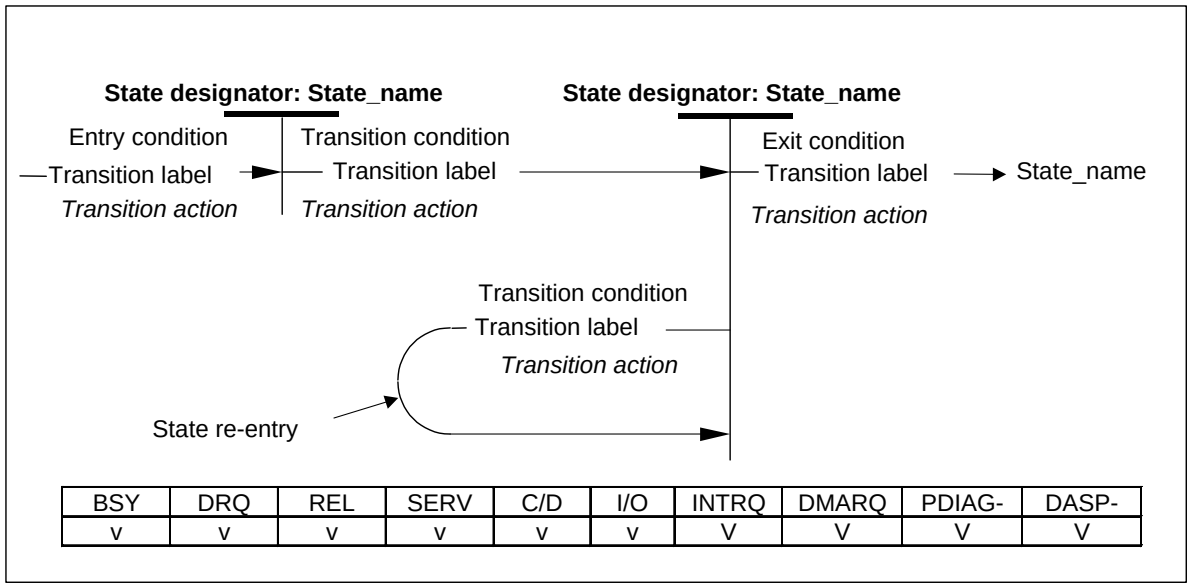



**Figure 2** – **State diagram convention**

Each state is identified by a state designator and a state name. The state designator is unique among all states in all state diagrams in this document. The state designator consists of a set of letters that are capitalized in the title of the figure containing the state diagram followed by a unique number. The state name is a brief description of the primary action taken during the state, and the same state name may appear in other state diagrams. If the same primary function occurs in other states in the same state diagram, they are designated with a unique letter at the end of the name. Additional actions may be taken while in a state and these actions are described in the state description text.

In device command protocol state diagrams, the state of bits and signals that change state during the execution of this state diagram are shown under the state designator:state_name, and a table is included that shows the state of all bits and signals throughout the state diagram as follows:

v = bit value changes.
1 = bit set to one.
0 = bit cleared to zero.
x = bit is don't care.
V = signal changes.
A = signal is asserted.
N = signal is negated.
R = signal is released.
X = signal is don't care.

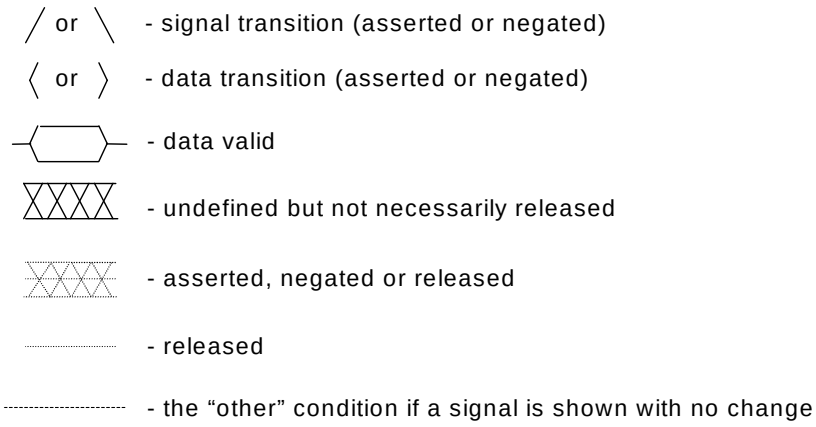
Each transition is identified by a transition label and a transition condition. The transition label consists of the state designator of the state from which the transition is being made followed by the state designator of the state to which the transition is being made. In some cases, the transition to enter or exit a state diagram may come from or go to a number of state diagrams, depending on the command being executed. In this case, the state designator is labeled xx. The transition condition is a brief description of the event or condition that causes the transition to occur and may include a transition action, indicated in italics, that is taken when the transition occurs. This action is described fully in the transition description text.

Upon entry to a state, all actions to be executed in that state are executed. If a state is re-entered from itself, all actions to be executed in the state are executed again.

Transitions from state to state shall be instantaneous.

### 3.2.8 Timing conventions

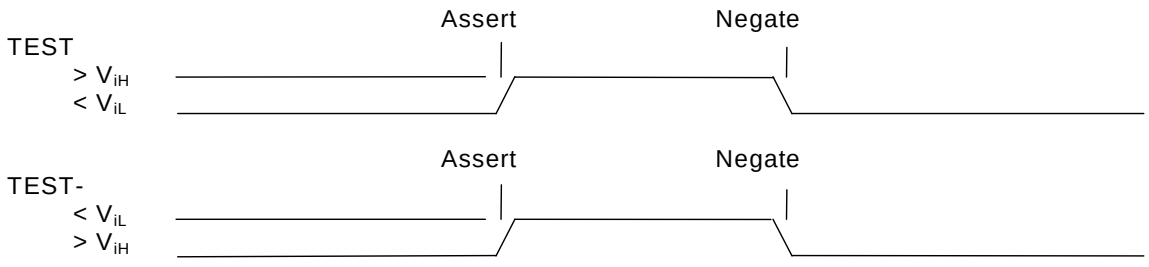
Certain symbols are used in the timing diagrams. These symbols and their respective definitions are listed below.




All signals are shown with the asserted condition facing to the top of the page. The negated condition is shown towards the bottom of the page relative to the asserted condition.

The interface uses a mixture of negative and positive signals for control and data. The terms asserted and negated are used for consistency and are independent of electrical characteristics.

In all timing diagrams, the lower line indicates negated, and the upper line indicates asserted. The following illustrates the representation of a signal named TEST going from negated to asserted and back to negated, based on the polarity of the signal.




### 3.2.9 Byte ordering for data transfers

Data is transferred in blocks using either PIO or DMA protocols. PIO data transfers occur when the BSY bit is cleared to zero and the DRQ bit is set to one. These transfers are usually 16-bit but CFA devices may

implement 8-bit PIO transfers. Data is transferred in blocks of one or more bytes known as a DRQ block. DMA data transfers occur when the host asserts DMACK- in response to the device asserting DMARQ. DMA transfers are always 16-bit. Each assertion of DMACK- by the host defines a DMA data burst. A DMA data burst is two or more bytes.

Assuming a DRQ block or a DMA burst of data contains "n" bytes of information, the bytes are labeled Byte(0) through Byte(n-1), where Byte(0) is first byte of the block, and Byte(n-1) is the last byte of the block. **Table 2** shows the order the bytes shall be presented in when such a block of data is transferred on the interface using 16-bit PIO and DMA transfers. **Table 3** shows the order the bytes shall be presented in when such a block or burst of data is transferred on the interface using 8-bit PIO.

**Table 2 – Byte order**

| | **DD 15** | **DD 14** | **DD 13** | **DD 12** | **DD 11** | **DD 10** | **DD 9** | **DD 8** | **DD 7** | **DD 6** | **DD 5** | **DD 4** | **DD 3** | **DD 2** | **DD 1** | **DD 0** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| First transfer | Byte (1) | | | | | | | | Byte (0) | | | | | | | |
| Second transfer | Byte (3) | | | | | | | | Byte (2) | | | | | | | |
| ..... | | | | | | | | | | | | | | | | |
| Last transfer | Byte (n-1) | | | | | | | | Byte (n-2) | | | | | | | |

**Table 3 – Byte order**

| | **DD 7** | **DD 6** | **DD 5** | **DD 4** | **DD 3** | **DD 2** | **DD 1** | **DD 0** |
|---|---|---|---|---|---|---|---|---|
| First transfer | Byte (0) | | | | | | | |
| Second transfer | Byte (1) | | | | | | | |
| ..... | | | | | | | | |
| Last transfer | Byte (n-1) | | | | | | | |

NOTE – The above description is for data on the interface. Host systems and/or host adapters may cause the order of data as seen in the memory of the host to be different.

Some parameters are defined as a string of ASCII characters. ASCII data fields shall contain only code values 20h through 7Eh. For the string “Copyright”, the character “C” is the first byte, the character “o” is the second byte, etc. When such fields are transferred, the order of transmission is:

the 1st character (“C”) is on DD(15:8) of the first word,
the 2nd character (“o”) is on DD(7:0) of the first word,
the 3rd character (“p”) is on DD(15:8) of the second word,
the 4th character (“y”) is on DD(7:0) of the second word,
the 5th character (“r”) is on DD(15:8) of the third word,
the 6th character (“i”) is on DD(7:0) of the third word,
the 7th character (“g”) is on DD(15:8) of the fourth word,
the 8th character (“h”) is on DD(7:0) of the fourth word,
the 9th character (“t”) is on DD(15:8) of the fifth word,
the 10th character (“space”) is on DD(7:0) of the fifth word,
etc.

Word (n:m) denotes a set of words, for example, words (103:100).

### 3.2.10 Byte, word and DWORD Relationships

**Figure 3** illustrates the relationship between bytes, words and DWORDs for serial interface implementations.

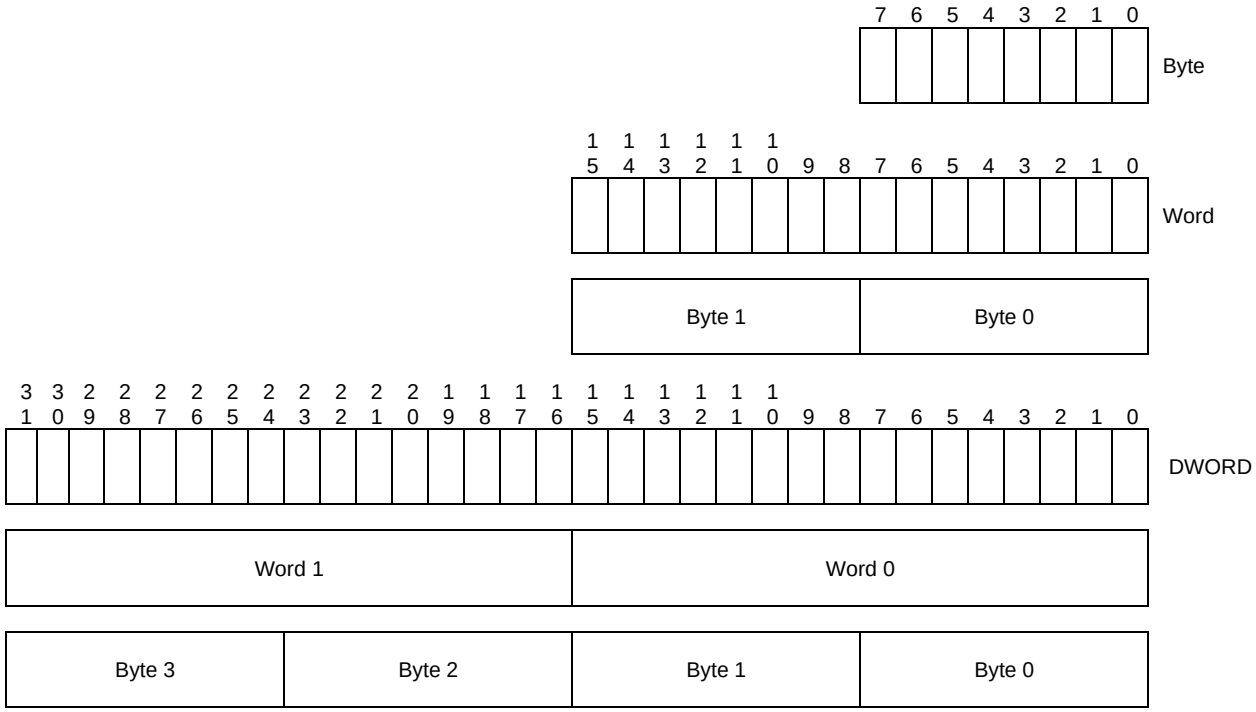



**Figure 3 - Byte, word and DWORD relationships**

**4 General command operational requirements** (See Volume 1)

5 **Interface register descriptions** (See Volume 1)
(Also see clause 10)

**6 Command descriptions** (See Volume 1)

# 7 Parallel interface physical and electrical requirements

## 7.1 Cable configuration

This standard defines an interface containing a single host or host adapter and one or two devices. One device is configured as Device 0 and the other device as Device 1.

The designation of a device as Device 0 or Device 1 may be made in a number of ways including but not limited to:

- a switch or a jumper on the device;
- use of the Cable Select (CSEL) pin.

The host shall be placed at one end of the cable. It is recommended that for a single device configuration the device be placed at the opposite end of the cable from the host. If a single device configuration is implemented with the device not at the end of the cable, a cable stub results that may cause degradation of signals. Single device configurations with the device not at the end of the cable shall not be used with Ultra DMA modes.

## 7.2 Electrical characteristics

Table 4 defines the DC characteristics of the interface signals. Table 5 defines the AC characteristics. These characteristics apply to both host and device unless otherwise specified.

**Table 4 – DC characteristics**

| Description | | Min | Max |
|---|---|---|---|
| $I_{oL}$ | Driver sink current (see note 1) | 4 mA | |
| $I_{oLDASP}$ | Driver sink current for DASP (see note 1) | 12 mA | |
| $I_{oH}$ | Driver source current (see note 2) | 400 µA | |
| $I_{oHDMARQ}$ | Driver source current for DMARQ (see note 2) | 500 µA | |
| $I_Z$ | Device pull-up current on DD(15:8), DD(6:0), and STROBE when released | -100 µA | 200 µA |
| $I_{ZDD7}$ | Device pull-up current on DD7 when released | -100 µA | 10 µA |
| $V_{iH}$ | Voltage input high | 2.0 VDC | 5.5 VDC |
| $V_{iL}$ | Voltage input low | | 0.8 VDC |
| $V_{oH}$ | Voltage output high at $I_{oH}$ min (see note 3) | 2.4 VDC | |
| $V_{oL}$ | Voltage output low at $I_{oL}$ min (see note 3) | | 0.5 VDC |
| **Additional DC characteristics for Ultra DMA modes greater than 4** | | | |
| $V_{DD3}$ | DC supply voltage to drivers and receivers | 3.3 V - 8% | 3.3 V + 8% |
| V+ | Low to high input threshold | 1.5 V | 2.0 V |
| V– | High to low input threshold | 1.0 V | 1.5 V |
| $V_{HYS}$ | Difference between input thresholds: $((V+_{current\ value}) - (V-_{current\ value}))$ | 320 mV | |
| $V_{THRAVG}$ | Average of thresholds: $((V+_{current\ value}) + (V-_{current\ value}))/2$ | 1.3 V | 1.7 V |
| $V_{oH2}$ | Voltage output high at -6 mA to +3 mA (at $V_{oH2}$ the output shall be able to supply and sink current to $V_{DD3}$) (see note 3) | $VDD_3$ - 0.51 VDC | $VDD_3$ + 0.3 VDC |
| $V_{oL2}$ | Voltage output low at 6 mA (see note 3) | | 0.51 VDC |

NOTES –

1 $I_{oLDASP}$ shall be 12 mA minimum to meet legacy timing and signal integrity.

2 $I_{oH}$ value at 400 µA is insufficient in the case of DMARQ that is pulled low by a 5.6 kΩ resistor.

3. Voltage output high and low values shall be met at the source connector to include the effect of series termination.

4. A device shall have less than 64 µA of leakage current into a 6.2 KΩ pull-down resistor while the INTRQ signal is in the released state.

**Table 5 – AC characteristics**

| | Description | Min | Max |
|---|---|---|---|
| $S_{RISE}$ | Rising edge slew rate for any signal (see note 1) | | 1.25 V/ns |
| $S_{FALL}$ | Falling edge slew rate for any signal (see note 1) | | 1.25 V/ns |
| $C_{host}$ | Host interface signal capacitance at the host connector (see note 2) | | 25 pf |
| $C_{device}$ | Device interface signal capacitance at the device connector (see note 2) | | 20 pf |
| **Additional AC characteristics for Ultra DMA modes greater than mode 4** | | | |
| $S_{RISE2}$ | Rising edge slew rate for DD(15:0) and STROBE (see note 1) | 0.40 V/ns | 1.0 V/ns |
| $S_{FALL2}$ | Falling edge slew rate for DD(15:0) and STROBE (see note 1) | 0.40 V/ns | 1.0 V/ns |
| $V_{DSSOH}$ | Induced signal to conductor side of device connector for any non-switching data signal at $V_{oH}$ due to simultaneous switching of all other data lines high and low by the device (see note 3) | $V_{DD3}$ - 500 mV | |
| $V_{DSSOL}$ | Same as $V_{DSSOH}$ except non-switching data signal at $V_{oL}$ (see note 3) | | 500 mV |
| $V_{HSSOH}$ | Induced signal to conductor side of host connector for any non-switching data signal at $V_{oH}$ due to simultaneous switching of all other data lines high and low by the host (see note 3) | $V_{DD3}$ - 600 mV | |
| $V_{HSSOL}$ | Same as $V_{HSSOH}$ except non-switching data signal at $V_{oL}$ (see note 3) | | 600 mV |
| $V_{RING}$ | AC voltage at recipient connector (see note 4) | -1.0 V | 6.0 V |
| $C_{device2}$ | Device capacitance measured at the connector pin (see note 2) | | 17 pf |
| $C_{ratio}$ | Ratio of the highest DD(15:0) or STROBE signal capacitance as measured at the connector to the lowest DD(15:0) or STROBE signal capacitance. | | 1.5 |
| $V_{ihPEAK}$ | The highest voltage reached on a rising transition at the recipient connector within 3 ns of crossing 1.5 V (see note 5) | 2.2 V | |
| $V_{ihRING}$ | The lowest voltage on a high signal at the recipient connector at any time after the rising edge crosses $V_{ihPEAK}$ until activity driven low by a subsequent falling transition (see note 5) | 1.7 V | |
| $V_{ilPEAK}$ | The lowest voltage reached on a falling transition at the recipient connector within 3 ns of crossing 1.5 V (see note 5) | | 0.8 V |
| $V_{ilRING}$ | The highest voltage on a low signal at the recipient connector at any time after the falling edge crosses $V_{ilPEAK}$ until activity driven high by a subsequent rising transition (see note 5) | | 1.3 V |

NOTES -
1 Signal integrity may be improved by using slower slew rates at slower transfer rates.
2 Capacitance measured at 1 MHz.
3 See 7.2.1.2 for measurement details.
4 The sender shall not generate voltage peaks higher then these absolute limits on DD(15:0) with all data lines switching simultaneously and a single recipient at end of cable. The test load shall be an 18" long, 40-conductor cable in Ultra DMA mode 2, as well as, an 18", long 80-conductor cable operated in the highest Ultra DMA mode supported.
5 $V_{ihPEAK}$ , $V_{ihRING}$ , $V_{ilPEAK}$ , and $V_{ilRING}$ shall be met in a functioning system across all patterns and shall be met when measured at any connector.

### 7.2.1 AC characteristics measurement techniques

#### 7.2.1.1 Slew rate

The sender's signals shall be tested while driving an 18" long, 80-conductor cable with PVC insulation material. The signal under test shall be cut at a test point so that it has no trace, cable, or recipient loading after the test point. All other signals should remain connected through to the recipient. The test point may be located at any point between the sender's series termination resistor and 0.5" or less of conductor exiting the connector. If the test point is on a cable conductor rather than the PCB, an adjacent ground conductor shall also be cut within 0.5" of the connector. The test load and test points should then be soldered directly to the exposed source side connectors. The test load shall consist of a 15 or 40 pf, 5%, 0.08" by 0.05", surface mount (or smaller size) capacitor from the test point to ground. Slew rates shall be met for both capacitor

values. Measurements shall be taken at the test point using a 1 GHz or faster test probe with less than 1 pf capacitance and greater than 100 kΩ impedance connected to a 500 MHz or faster oscilloscope. The average rate shall be measured from 20% to 80% of the settled $V_{oH}$ level with data transitions at least 120 ns apart. The settled $V_{oH}$ level shall be measured as the average output high level under the defined testing conditions from 100 ns after 80% of a rising edge until 20% of the subsequent falling edge.

#### 7.2.1.2 $V_{SSO}$

$V_{sso}$ shall be tested with the same test cable configuration as described for slew rate testing except with the test load described here and the cut-cable-conductor configuration. For both $V_{oL}$ and $V_{oH}$ measurements, the test load shall consist of a 90.9 Ω 1% resistor (this also may be accomplished by using 1 kΩ 1% and 100 Ω 1% resistors in parallel) and a 0.1 µf 20% capacitor in series to ground. Both resistor and capacitor shall be 0.08" by 0.05" or smaller surface mount. The order of components should be signal-resistor-capacitor-ground. Refer to 7.2.3.3 for PCB layout requirements related to $V_{SSO}$.

To generate a test pattern for $V_{SSOH}$ a 32-sector or longer data transfer shall be sent to the recipient. The first 30 sectors shall contain Fh in order to generate an "all 1's" pattern to pre-charge the capacitor to $V_{oH}$. The final sectors of the command shall contain the required pattern for $V_{SSOH}$ for at least 2 sectors (i.e., a pattern that holds the data line under test at $V_{oH}$ while driving a "1010…" pattern on all other data lines. $V_{SSOH}$ shall be measured within the first 32 words after the pre-charge pattern is complete and the $V_{SSOH}$ pattern begins.

To generate a test pattern for $V_{SSOL}$ a 32-sector or longer data transfer shall be sent to the recipient. The first 30 sectors shall contain 0h in order to generate an "all 0's" pattern to pre-charge the capacitor to $V_{oL}$. The final sectors of the command shall contain the required pattern for $V_{SSOL}$ for at least 2 sectors (i.e., a pattern that holds the data line under test at $V_{oL}$ while driving a "1010…" pattern on all other data lines. $V_{SSOL}$ shall be measured within the first 32 words after the pre-charge pattern is complete and the $V_{SSOL}$ pattern begins.

### 7.2.2 Driver types and required termination

**Table 6 – Driver types and required termination**

| Signal | Source | Driver type (see note 1) | Host (see note 2) | Device (see note 2) | Notes |
|---|---|---|---|---|---|
| RESET- | Host | TP | | | |
| DD(15:0) | Bidir | TS | | | 3 |
| DMARQ | Device | TS | 5.6 kΩ PD | | |
| DIOR-:HDMARDY-:HSTROBE | Host | TS | | | |
| DIOW-:STOP | Host | TS | | | |
| IORDY:DDMARDY-:DSTROBE | Device | TS | 4.7 kΩ PU | | 6,10 |
| CSEL | Host | | Ground | 10 kΩ PU | 4, 6 |
| DMACK- | Host | TP | | | |
| INTRQ | Device | TS | 10 kΩ | | 5 |
| DA(2:0) | Host | TP | | | |
| PDIAG-:CBLID- | Device | TS | | 10 kΩ PU | 2,6,7,8 |
| CS0- CS1- | Host | TP | | | |
| DASP- | Device | OC | | 10 kΩ PU | 6,9 |

NOTES –

1 TS=Tri-state; OC=Open Collector; TP=Totem-pole; PU=Pull-up; PD=Pull-down.

2 All resistor values are the minimum (lowest allowed) except for the 10 kΩ PU on PDIAG-:CBLID- which shall have a tolerance of ±5% or less.

3 Devices shall not have a pull-up resistor on DD7. The host shall have a 10 kΩ pull-down resistor and not a pull-up resistor on DD7 to allow a host to recognize the absence of a device at power-up so that a host shall detect BSY as being cleared when attempting to read the Status register of a device that is not present.

4 When used as CSEL, this line is grounded at the host and a 10 kΩ pull-up is required at both devices.

5 A 10 kΩ pull-up or 6.2 kΩ pull-down, depending upon the level sensed, should be implemented at the host.

6 Pull-up values are based on +5 V Vcc. Except for the pull-up on PDIAG-:CBLID- which shall be to +5 $V_{CC}$ for backward compatibility, pull-ups may be to $V_{DD3}$. For systems supporting Ultra DMA modes greater than 4, the host pull-up on IORDY:DDMARDY-:DSTROBE should be to $V_{DD3}$.

7 Hosts that do not support Ultra DMA modes greater than mode 2 shall not connect to the PDIAG-:CBLID- signal.

8 The 80-conductor cable assembly shall meet the following requirements: the PDIAG-:CBLID- signal shall be connected to ground in the host connector of the cable assembly; the PDIAG-:CBLID- signal shall not be connected between the host and the devices; and, the PDIAG-:CBLID- signal shall be connected between the devices.

9 The host shall not drive DASP-. If the host connects to DASP- for any purpose, the host shall ensure that the signal level detected on the interface for DASP- shall maintain $V_{oH}$ and $V_{oL}$ compatibility, given the $I_{oH}$ and $I_{oL}$ requirements of the DASP- device drivers.

10 For host systems not supporting modes greater than Ultra DMA mode 4, a pull-up of 1 kΩ may be used.

### 7.2.3 Electrical characteristics for Ultra DMA

Hosts that support Ultra DMA transfer modes greater than mode 2 shall not share signals between primary and secondary I/O ports. They shall provide separate drivers and separate receivers for each cable.

#### 7.2.3.1 Cable configuration

The following table defines the host transceiver configurations for a dual cable system configuration for all transfer modes.

| Transfer mode | Optional host transceiver configuration | Recommended host transceiver configuration | Mandatory host transceiver configuration |
|---|---|---|---|
| All PIO and Multiword DMA | One transceiver may be used for signals to both ports. | DIOR-, DIOW-, and IORDY should have a separate transceiver for each port. | Either DIOR-, DIOW-, and IORDY or CS0- and CS1- shall have a separate transceiver for each port. |
| Ultra DMA 0, 1, 2 | One transceiver may be used for signals to both ports except DMACK-. | DIOR-, DIOW-, and IORDY should have a separate transceiver for each port. | Either DIOR-, DIOW-, and IORDY or CS0- and CS1- shall have a separate transceiver for each port. DMACK- shall have a separate transceiver for each port. |
| Ultra DMA modes > 2 | One transceiver may be used for signals to both ports for RESET-, INTRQ, DA(2:0), CS0-, CS1-, and DASP-. | RESET-, INTRQ, DA(2:0), CS0-, CS1-, and DASP- should have a separate transceiver for each port. | All signals shall have a separate transceiver for each port except for RESET-, INTRQ, DA(2:0), CS0-, CS1-, and DASP-. |

The following table defines the system configuration for connection between devices and systems for all transfer modes. For Ultra DMA modes requiring an 80-conductor cable, that cable shall meet the requirements for 80-conductor cables(see 7.3.1.2).

| Transfer mode | Single device direct connection configuration (see note 1) | 40-conductor cable connection configuration (see note 2) | 80-conductor cable connection configuration (see note 2) |
|---|---|---|---|
| All PIO and Multiword DMA | May be used. | May be used. | May be used (see note 3) |
| Ultra DMA 0, 1, 2 | May be used. | May be used. | May be used (see note 3) |
| Ultra DMA modes > 2 | May be used (see note 4). | Shall not be used. | May be used (see note 4). |

NOTES -
1 Direct connection is a direct point-to-point connection between the host connector and the device connector.
2 The 40-conductor cable assembly and the 80-conductor cable assembly are defined in 7.3.
3 80-conductor cable assemblies may be used in place of 40-conductor cable assemblies to improve signal quality for data transfer modes that do not require an 80-conductor cable assembly.
4 Either a single device direct connection configuration or an 80-conductor cable connection configuration shall be used for systems operating with Ultra DMA modes greater than 2.

#### 7.2.3.2 Series termination required for Ultra DMA

Series termination resistors are required at both the host and the device for operation in any of the Ultra DMA modes. Table 7 describes typical values for series termination at the host and the device.

For host systems and devices supporting Ultra DMA modes greater than 4, the output and bi-directional series termination values for DD(15:0) and STROBE signals shall be chosen so that the sum of the driver output resistance at $V_{oL2}$ or $V_{oH2}$ and the series termination resistance is between 50 and 85 Ω. For these systems, the STROBE input shall use the same series termination resistance value as the data lines.

Host systems supporting Ultra DMA modes greater than 5 and having PCB traces longer than 4" shall use series termination resistors of no less than 22 Ω on DD(15:0) and STROBE signals. The termination resistors shall be placed within 0.5" of the host connector.

**Table 7 – Typical series termination for Ultra DMA**

| Signal | Host Termination | Device Termination |
|---|---|---|
| DIOR-:HDMARDY-:HSTROBE | 22 ohm | 82 ohm |
| DIOW-:STOP | 22 ohm | 82 ohm |
| CS0-, CS1- | 33 ohm | 82 ohm |
| DA0, DA1, DA2 | 33 ohm | 82 ohm |
| DMACK- | 22 ohm | 82 ohm |
| DD15 through DD0 | 33 ohm | 33 ohm |
| DMARQ | 82 ohm | 22 ohm |
| INTRQ | 82 ohm | 22 ohm |
| IORDY:DDMARDY-:DSTROBE | 82 ohm | 22 ohm |
| RESET- | 33 ohm | 82 ohm |

NOTE – Only those signals requiring termination are listed in this table. If a signal is not listed, series termination is not required for operation in an Ultra DMA mode. Figure 4 shows signals also requiring a pull-up or pull-down resistor at the host. The actual termination values should be selected to compensate for transceiver and trace impedance to match the characteristic cable impedance.

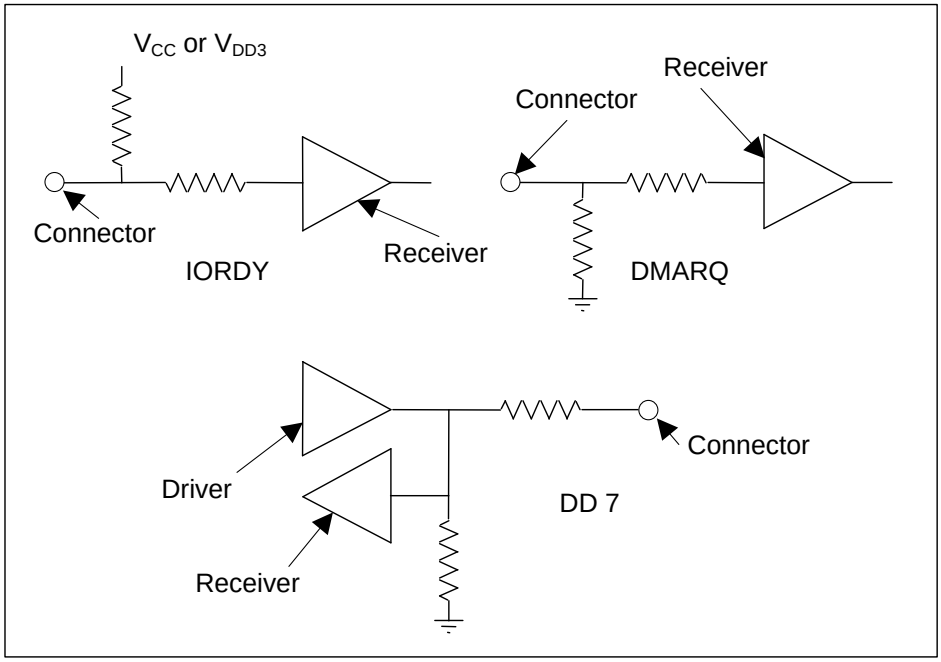



**Figure 4 – Ultra DMA termination with pull-up or pull-down**

#### 7.2.3.3 PCB trace requirements for Ultra DMA

PCB trace layout is a factor in meeting the $V_{sso}$ values in table 4.

The longest DD(15:0) or STROBE trace for any device supporting Ultra DMA modes greater than 5 shall be 4".

On any PCB for a host or device supporting Ultra DMA: The longest DD(15:0) trace shall be no more than 0.5" longer than either STROBE trace as measured from the IC pin to the connector. The shortest DD(15:0) trace shall be no more than 0.5" shorter than either STROBE trace as measured from the IC pin to the connector.

Any DD(15:0) or STROBE trace on a PCB for a host or device supporting Ultra DMA modes greater than 5 shall meet the following:

$$L / (1 + (D / H)^2 ) < 0.8$$

Where:

L is the trace length in inches
D is the center-to-center trace spacing of adjacent traces. If both traces are the same width, this is equivalent to one trace width plus the trace separation.
H is the height of the trace above a continuous, unbroken supply plane. The plane may be power or ground but if a power plane is used, it must be bypassed to the ground plane on both sides of the DD(15:0) bus near the connector ground pins and at the IC.

The units used for D and H shall be the same.

## 7.3 Connectors and cable asemblies

The device shall implement one of the connector options described in this clause.

### 7.3.1 40-pin Connector

The I/O connector is a 40-pin connector. The header mounted to a host or device is shown in Figure 5 and the dimensions are shown in Table 8. The connector mounted to the end of the cable is shown in Figure 6 and the dimensions are shown in Table 9. Signal assignments on these connectors are shown in Table 10.

The pin locations are governed by the cable plug, not the receptacle. The way in which the receptacle is mounted on the printed circuit board affects the pin positions, and pin 1 shall remain in the same relative position. This means the pin numbers of the receptacle may not reflect the conductor number of the plug. The header receptacle may or may not be polarized, and all the signals are relative to pin 20, which is keyed.

By using the plug positions as primary, a straight cable can connect devices. As shown in Figure 7, conductor 1 on pin 1 of the plug shall be in the same relative position no matter what the receptacle numbering looks like. If receptacle numbering was followed, the cable would have to twist 180 degrees between a device with top-mounted receptacles, and a device with bottom-mounted receptacles.

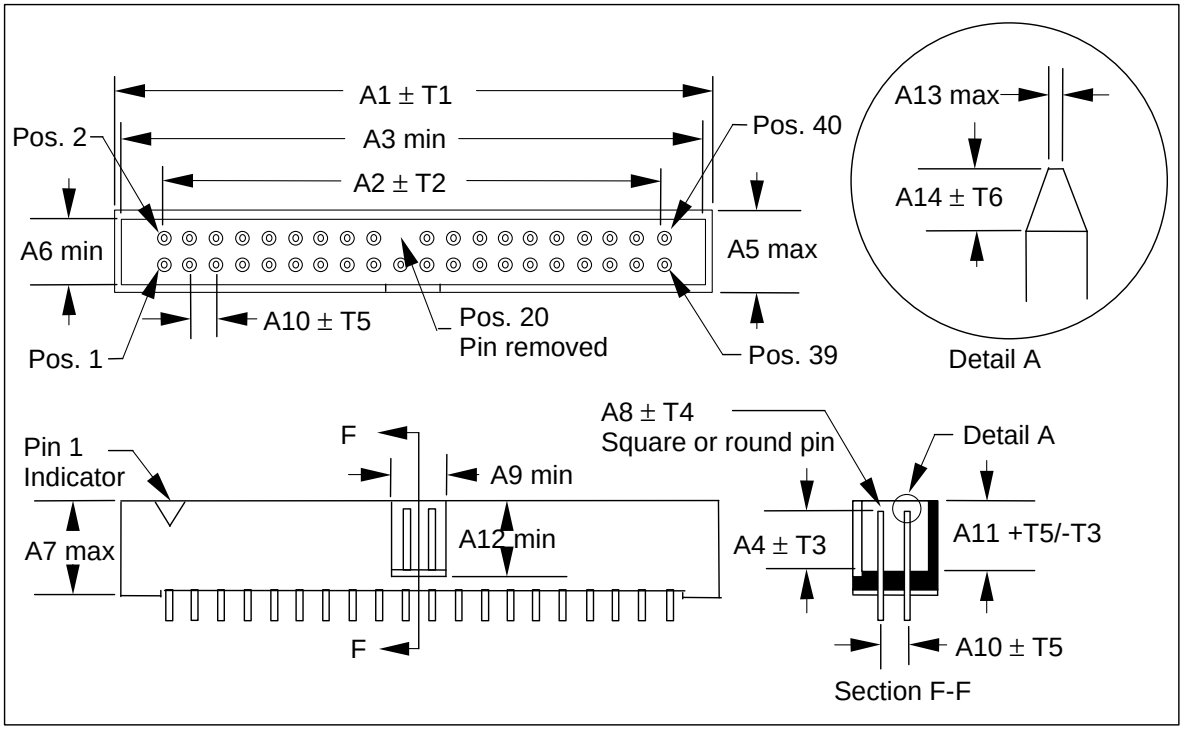



**Figure 5 – Host or device 40-pin I/O header**

**Table 8 – Host or device 40-pin I/O header**

| Dimension | Millimeters | Inches |
|---|---|---|
| A 1 | 58.17 | 2.290 |
| A 2 | 48.26 | 1.900 |
| A 3 | 56.01 | 2.205 |
| A 4 | 5.84 | 0.230 |
| A 5 | 9.55 | 0.376 |
| A 6 | 6.22 | 0.245 |
| A 7 | 10.16 | 0.400 |
| A 8 | 0.64 | 0.025 |
| A 9 | 4.06 | 0.160 |
| A 10 | 2.54 | 0.100 |
| A 11 | 6.35 | 0.250 |
| A 12 | 6.48 | 0.255 |
| A 13 | 0.33 | 0.013 |
| A 14 | 0.58 | 0.023 |
| T 1 | 0.51 | 0.020 |
| T 2 | 0.13 | 0.005 |
| T 3 | 0.25 | 0.010 |
| T 4 | 0.03 | 0.001 |
| T 5 | 0.08 | 0.003 |
| T 6 | 0.18 | 0.007 |

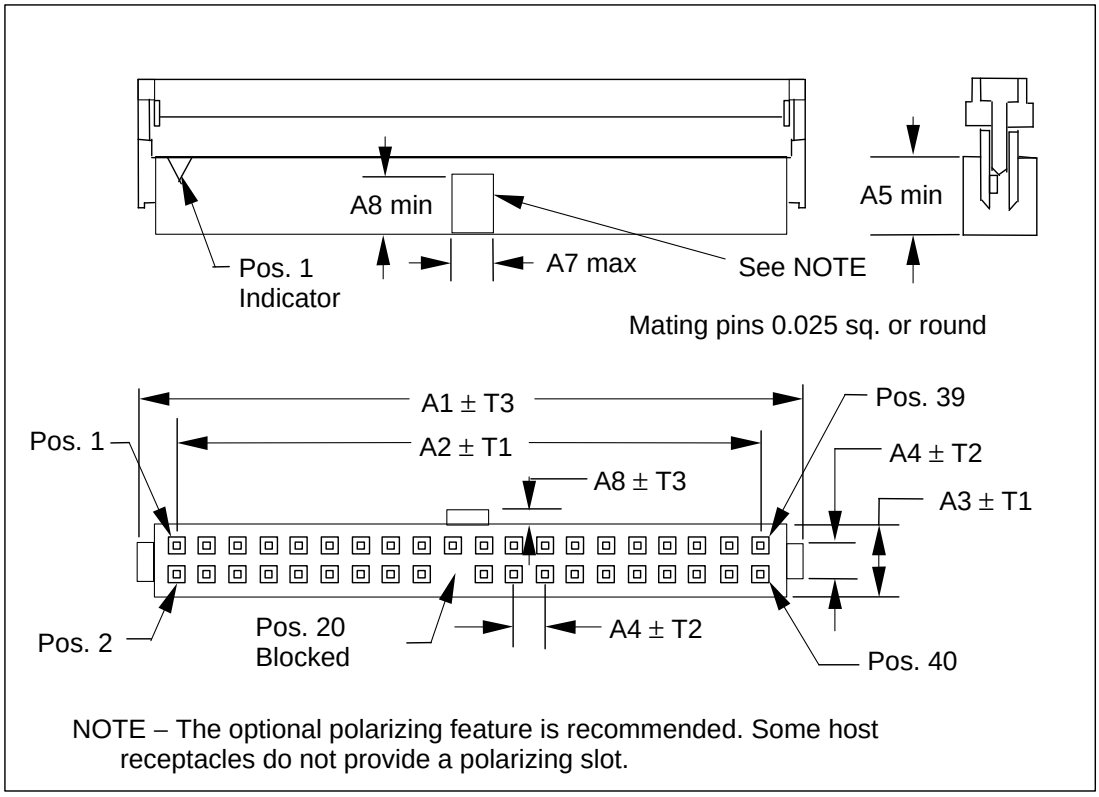



**Figure 6 – 40-pin I/O cable connector**

**Table 9 – 40-pin I/O cable connector**

| Dimension | Millimeters | Inches |
|---|---|---|
| A 1 | 55.37 | 2.180 |
| A 2 | 48.26 | 1.900 |
| A 3 | 6.10 | 0.240 |
| A 4 | 2.54 | 0.100 |
| A 5 | 6.48 | 0.255 |
| A 6 | 4.57 | 0.180 |
| A 7 | 3.81 | 0.150 |
| A 8 | 1.27 | 0.050 |
| T 1 | 0.13 | 0.005 |
| T 2 | 0.08 | 0.003 |
| T 3 | 0.25 | 0.010 |

**Table 10 – 40-pin I/O connector interface signals**

| Signal name | Connector contact | Conductor | | Connector contact | Signal name |
|---|---|---|---|---|---|
| RESET- | 1 | 1 | 2 | 2 | Ground |
| DD7 | 3 | 3 | 4 | 4 | DD8 |
| DD6 | 5 | 5 | 6 | 6 | DD9 |
| DD5 | 7 | 7 | 8 | 8 | DD10 |
| DD4 | 9 | 9 | 10 | 10 | DD11 |
| DD3 | 11 | 11 | 12 | 12 | DD12 |
| DD2 | 13 | 13 | 14 | 14 | DD13 |
| DD1 | 15 | 15 | 16 | 16 | DD14 |
| DD0 | 17 | 17 | 18 | 18 | DD15 |
| Ground | 19 | 19 | 20 | 20 | (keypin) |
| DMARQ | 21 | 21 | 22 | 22 | Ground |
| DIOW-:STOP | 23 | 23 | 24 | 24 | Ground |
| DIOR-:HDMARDY-:HSTROBE | 25 | 25 | 26 | 26 | Ground |
| IORDY:DDMARDY-:DSTROBE | 27 | 27 | 28 | 28 | CSEL |
| DMACK- | 29 | 29 | 30 | 30 | Ground |
| INTRQ | 31 | 31 | 32 | 32 | Obsolete (see note) |
| DA1 | 33 | 33 | 34 | 34 | PDIAG-:CBLID- |
| DA0 | 35 | 35 | 36 | 36 | DA2 |
| CS0- | 37 | 37 | 38 | 38 | CS1- |
| DASP- | 39 | 39 | 40 | 40 | Ground |
| NOTE – Pin 32 was defined as IOCS16 in ATA-2, ANSI X3.279-1996. | | | | | |

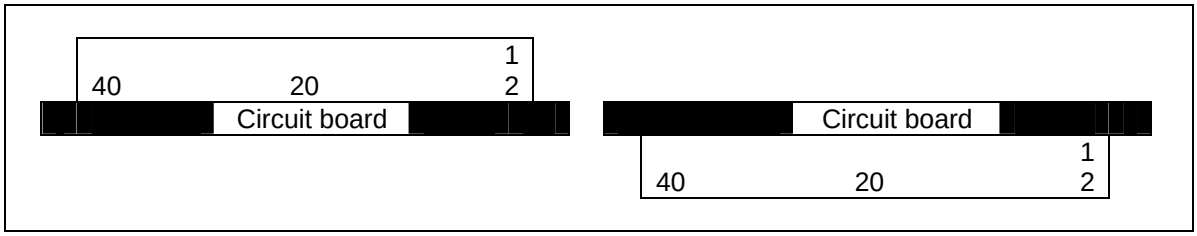



**Figure 7 – 40-pin I/O header mounting**

#### 7.3.1.1 40-conductor cable

The 40-conductor cable assemby is shown in Figure 8 with dimensions in Table 11. Cable capacitance shall not exceed 35 pf.

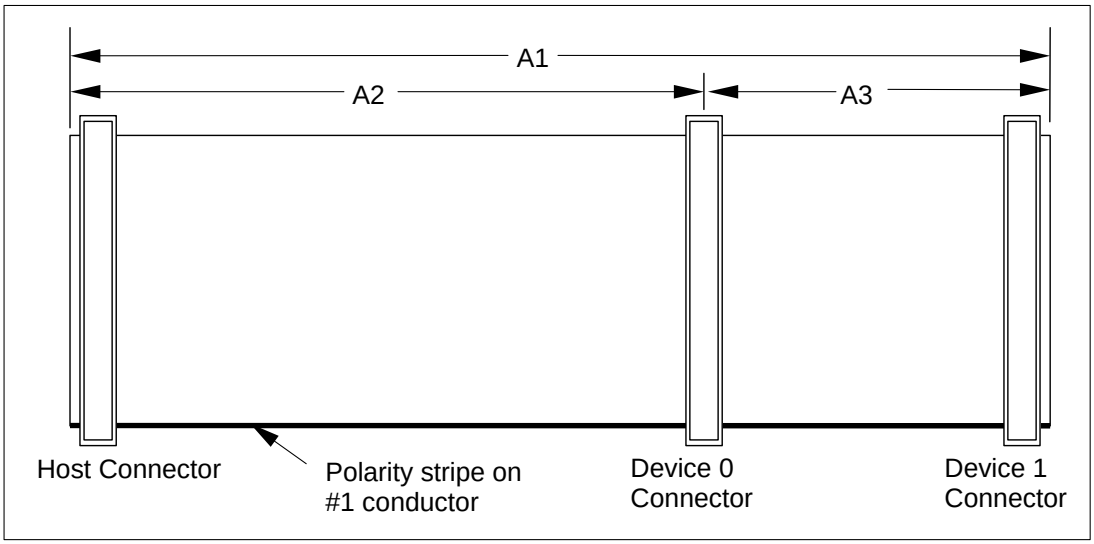



**Figure 8 – 40-conductor cable configuration**

**Table 11 – 40-conductor cable configuration**

| Dimension | Millimeters | Inches |
|---|---|---|
| A 1 | 254.00 min<br>457.20 max | 10.00 min<br>18.00 max |
| A 2 | 127.00 min<br>304.80 max | 5.00 min<br>12.00 max |
| A 3 | 127.00 min<br>152.40 max | 5.00 min<br>6.00 max |

#### 7.3.1.2 80-conductor cable assembly using the 40-pin connector

To provide better signal integrity, the optional 80-conductor cable assembly is specified for use with 40-pin connectors. Use of this assembly is mandatory for systems operating at Ultra DMA modes greater than 2. The mating half of the connector is as described in 7.3.1. Every other conductor in the 80-conductor cable is connected to the ground pins in each connector.

The electrical requirements of the 80-conductor ribbon cable are shown in Table 12 and the physical requirements are described in Figure 9 and Table 13.

Figure 10 and Table 14 describe the physical dimensions of the cable assembly. The connector in the center of the cable assembly labeled Device 1 Connector is optional. The System Board connector shall have a blue base and a black or blue retainer. The Device 0 Connector shall have a black base and a black retainer. The Device 1 Connector shall have a gray base and a black or gray retainer. The cable assembly may be printed with connector identifiers.

There are alternative cable conductor to connector pin assignments depending on whether the connector attaches all even or odd conductors to ground. Table 15 shows the signal assignments for connectors that ground the even numbered conductors. Table 16 shows the signal assignments for connectors that ground the odd numbered conductors. Only one connector type, even or odd, shall be used in a given cable assembly. Connectors shall be labeled as grounding the even or odd conductors as shown in Figure 11. Cable assemblies conforming to Table 15 are interchangable with cable assemblies conforming to Table 16.

All connectors shall have position 20 blocked to provide keying. Pin 28 in Device 1 Connector shall not be attached to any cable conductor, the connector contact may be removed to meet this requirement (see 8.2.13.2). Pin 34 in the Host Connector shall not be attached to any cable conductor and shall be attached to Ground within the connector (see 9.4).

**Table 12 – 80-conductor cable electrical requirements**

| | | |
|---|---|---|
| Conductor | | 30 AWG |
| Ground-signal-ground<br>Single ended impedance<br>Capacitance | <br>(Ω)<br>(pF/ft)<br>(pF/m) | <br>70-90<br>13-22<br>42-72 |
| Inductance | (μH) | 0.08-0.16 |
| Propagation delay | (nsec/ft)<br>(nsec/m) | 1.35-1.65<br>4.43-5.41 |

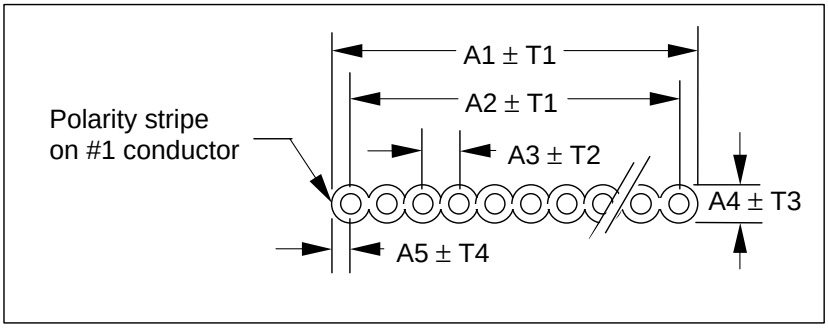



**Figure 9 – 80-conductor ribbon cable**

**Table 13 – 80-conductor ribbon cable**

| Dimension | Millimeters | Inches |
|---|---|---|
| A 1 | 50.800 | 2.000 |
| A 2 | 50.165 | 1.975 |
| A 3 | 0.635 | 0.025 |
| A 4 | 0.6858 | 0.027 |
| A 5 | 0.3175 | 0.0125 |
| T 1 | 0.127 | 0.005 |
| T 2 | 0.0406 | 0.0016 |
| T 3 | 0.0508 | 0.002 |
| T 4 | 0.102 | 0.004 |

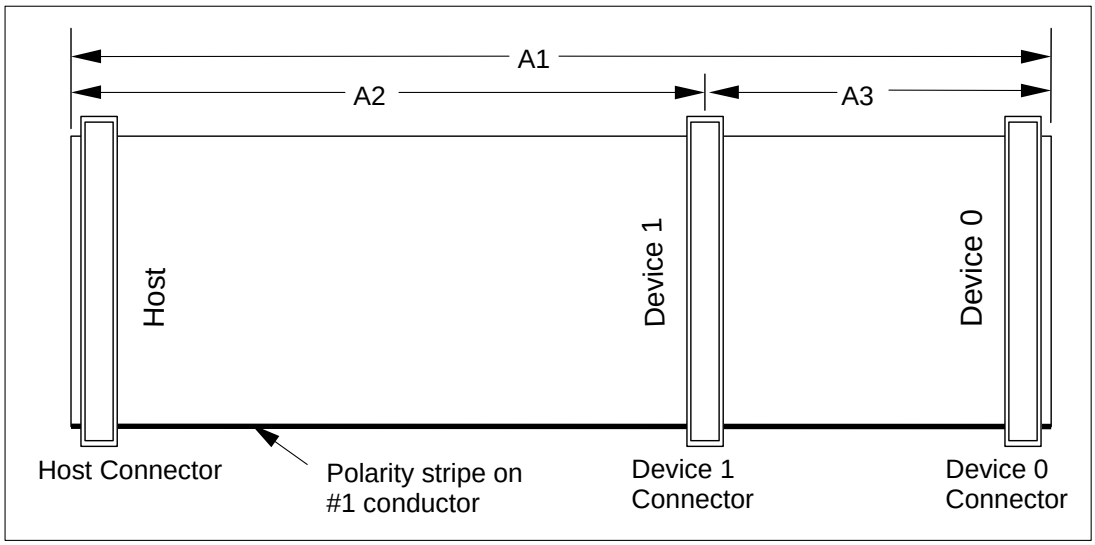



**Figure 10 – 80-conductor cable configuration**

**Table 14 – 80-conductor cable configuration**

| Dimension | Millimeters | Inches |
|---|---|---|
| A 1 | 457.20 max | 18.00 max |
| A 2 | 127.00 min | 5.00 min |
| A 3 | 152.40 max | 6.00 max |
| A2 min shall be greater than or equal to A3. | | |

**Table 15 – Signal assignments for connectors grounding even conductors**

| Signal name | Connector contact | Conductor | | Signal name |
|---|---|---|---|---|
| RESET- | 1 | 1 | 2 | Ground |
| Ground | 2 | 3 | 4 | Ground |
| DD7 | 3 | 5 | 6 | Ground |
| DD8 | 4 | 7 | 8 | Ground |
| DD6 | 5 | 9 | 10 | Ground |
| DD9 | 6 | 11 | 12 | Ground |
| DD5 | 7 | 13 | 14 | Ground |
| DD10 | 8 | 15 | 16 | Ground |
| DD4 | 9 | 17 | 18 | Ground |
| DD11 | 10 | 19 | 20 | Ground |
| DD3 | 11 | 21 | 22 | Ground |
| DD12 | 12 | 23 | 24 | Ground |
| DD2 | 13 | 25 | 26 | Ground |
| DD13 | 14 | 27 | 28 | Ground |
| DD1 | 15 | 29 | 30 | Ground |
| DD14 | 16 | 31 | 32 | Ground |
| DD0 | 17 | 33 | 34 | Ground |
| DD15 | 18 | 35 | 36 | Ground |
| Ground | 19 | 37 | 38 | Ground |
| (keypin) | 20 | 39 | 40 | Ground |
| DMARQ | 21 | 41 | 42 | Ground |
| Ground | 22 | 43 | 44 | Ground |
| DIOW- | 23 | 45 | 46 | Ground |
| Ground | 24 | 47 | 48 | Ground |
| DIOR- | 25 | 49 | 50 | Ground |
| Ground | 26 | 51 | 52 | Ground |
| IORDY | 27 | 53 | 54 | Ground |
| CSEL | 28 | 55 | 56 | Ground |
| DMACK- | 29 | 57 | 58 | Ground |
| Ground | 30 | 59 | 60 | Ground |
| INTRQ | 31 | 61 | 62 | Ground |
| Reserved | 32 | 63 | 64 | Ground |
| DA1 | 33 | 65 | 66 | Ground |
| PDIAG- | 34(see note) | 67 | 68 | Ground |
| DA0 | 35 | 69 | 70 | Ground |
| DA2 | 36 | 71 | 72 | Ground |
| CS0- | 37 | 73 | 74 | Ground |
| CS1- | 38 | 75 | 76 | Ground |
| DASP- | 39 | 77 | 78 | Ground |
| Ground | 40 | 79 | 80 | Ground |

NOTE – Pin 34 in the Host Connector shall not be attached to any cable conductor and shall be attached to Ground within the connector (see 9.4).

**Table 16 – Signal assignments for connectors grounding odd conductors**

| Signal name | Conductor | | Connector contact | Signal name |
|---|---|---|---|---|
| Ground | 1 | 2 | 1 | RESET- |
| Ground | 3 | 4 | 2 | Ground |
| Ground | 5 | 6 | 3 | DD7 |
| Ground | 7 | 8 | 4 | DD8 |
| Ground | 9 | 10 | 5 | DD6 |
| Ground | 11 | 12 | 6 | DD9 |
| Ground | 13 | 14 | 7 | DD5 |
| Ground | 15 | 16 | 8 | DD10 |
| Ground | 17 | 18 | 9 | DD4 |
| Ground | 19 | 20 | 10 | DD11 |
| Ground | 21 | 22 | 11 | DD3 |
| Ground | 23 | 24 | 12 | DD12 |
| Ground | 25 | 26 | 13 | DD2 |
| Ground | 27 | 28 | 14 | DD13 |
| Ground | 29 | 30 | 15 | DD1 |
| Ground | 31 | 32 | 16 | DD14 |
| Ground | 33 | 34 | 17 | DD0 |
| Ground | 35 | 36 | 18 | DD15 |
| Ground | 37 | 38 | 19 | Ground |
| Ground | 39 | 40 | 20 | (keypin) |
| Ground | 41 | 42 | 21 | DMARQ |
| Ground | 43 | 44 | 22 | Ground |
| Ground | 45 | 46 | 23 | DIOW- |
| Ground | 47 | 48 | 24 | Ground |
| Ground | 49 | 50 | 25 | DIOR- |
| Ground | 51 | 52 | 26 | Ground |
| Ground | 53 | 54 | 27 | IORDY |
| Ground | 55 | 56 | 28 | CSEL |
| Ground | 57 | 58 | 29 | DMACK- |
| Ground | 59 | 60 | 30 | Ground |
| Ground | 61 | 62 | 31 | INTRQ |
| Ground | 63 | 64 | 32 | Reserved |
| Ground | 65 | 66 | 33 | DA1 |
| Ground | 67 | 68 | 34 (see note) | PDIAG- |
| Ground | 69 | 70 | 35 | DA0 |
| Ground | 71 | 72 | 36 | DA2 |
| Ground | 73 | 74 | 37 | CS0- |
| Ground | 75 | 76 | 38 | CS1- |
| Ground | 77 | 78 | 39 | DASP- |
| Ground | 79 | 80 | 40 | Ground |

NOTE – Pin 34 in the Host Connector shall not be attached to any cable conductor and shall be attached to Ground within the connector (see 9.4).

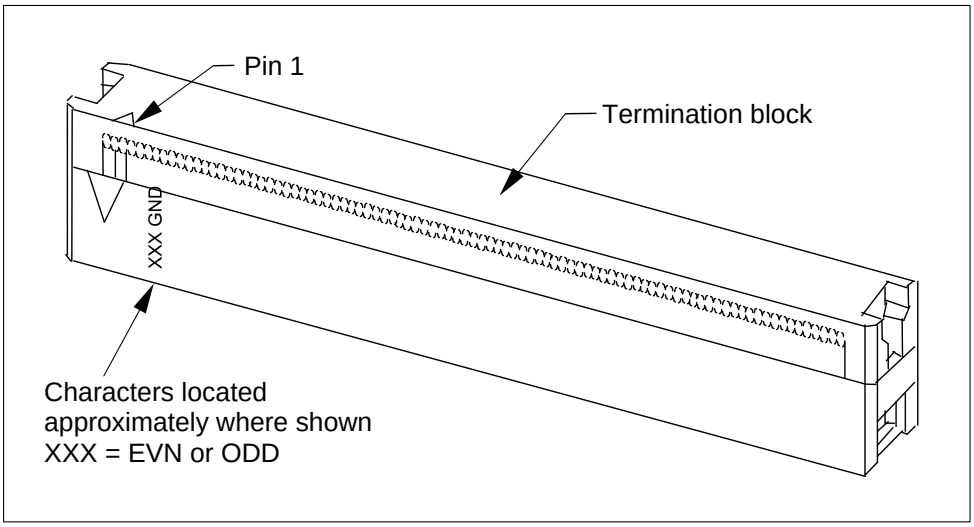
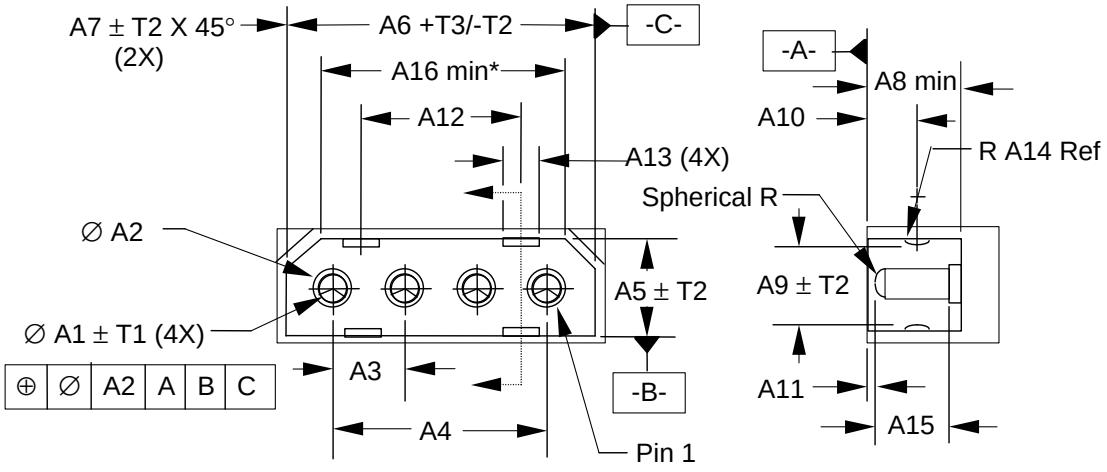



**Figure 11** – **Connector labeling for even or odd conductor grounding**

### 7.3.2 4-pin power connector

The power connector is a 4-pin connector. The header mounted to a device is shown in Figure 12 and the dimensions are shown in Table 17. The connector mounted to the end of the cable is shown in Figure 13 and the dimensions are shown in Table 18. Pin assignments for these connectors are shown in Table 19.




Unless otherwise specified, all tolerances are ± T3

* The tolerance build up of A6 and A7 shall not exceed A16

**Figure 12 – Device 4-pin power header**

**Table 17 – Device 4-pin power header**

| Dimension | Millimeters | Inches |
|---|---|---|
| A 1 | 2.10 | 0.083 |
| A 2 | 3.50 | 0.138 |
| A 3 | 5.08 | 0.200 |
| A 4 | 15.24 | 0.600 |
| A 5 | 6.60 | 0.260 |
| A 6 | 21.32 | 0.839 |
| A 7 | 1.65 | 0.065 |
| A 8 | 7.50 | 0.295 |
| A 9 | 6.00 | 0.236 |
| A 10 | 4.95 | 0.195 |
| A 11 | 1.00 | 0.039 |
| A 12 | 11.18 | 0.440 |
| A 13 | 3.80 | 0.150 |
| A 14 | 3.00 | 0.118 |
| A 15 | 5.10 | 0.201 |
| A 16 | 17.80 | 0.701 |
| T 1 | 0.04 | 0.0016 |
| T 2 | 0.15 | 0.006 |
| T 3 | 0.25 | 0.010 |

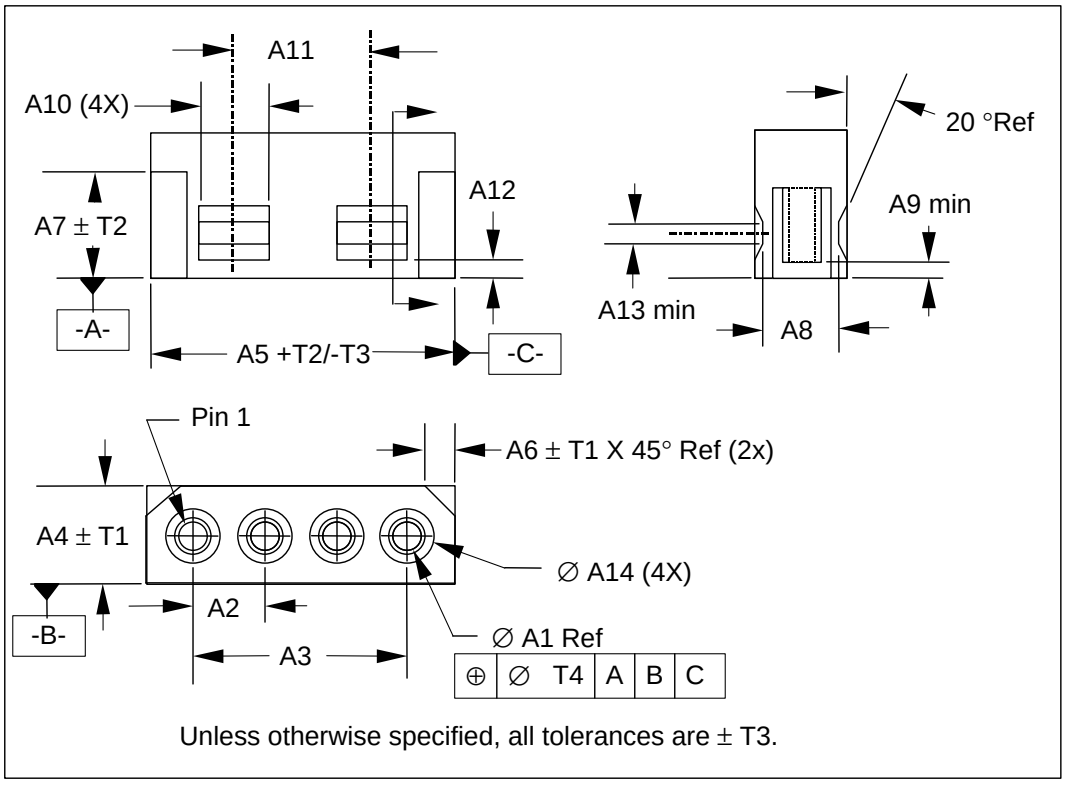



**Figure 13 – 4-pin power cable connector**

**Table 18 – 4-pin power cable connector**

| Dimension | Millimeters | Inches |
|---|---|---|
| A 1 | 2.03 | 0.080 |
| A 2 | 5.08 | 0.200 |
| A 3 | 15.24 | 0.600 |
| A 4 | 6.35 | 0.250 |
| A 5 | 21.00 | 0.827 |
| A 6 | 1.78 | 0.070 |
| A 7 | 7.87 | 0.310 |
| A 8 | 5.51 | 0.217 |
| A 9 | 1.19 | 0.047 |
| A 10 | 5.08 | 0.200 |
| A 11 | 11.18 | 0.440 |
| A 12 | 1.19 | 0.047 |
| A 13 | 2.00 | 0.079 |
| A 14 | 4.06 | 0.160 |
| T 1 | 0.10 | 0.004 |
| T 2 | 0.15 | 0.006 |
| T 3 | 0.25 | 0.010 |
| T 4 | 0.60 | 0.024 |

**Table 19 – 4-pin power connector pin assignments**

| Power line | Pin |
|---|---|
| +12 volts | 1 |
| +12 volt return | 2 |
| +5 volt return | 3 |
| +5 volts | 4 |

#### 7.3.2.1 Mating performance

Mating force should be 3.85 lbs (1.75 kg) maximum per contact.

Unmating force should be 0.25 lbs (113.5 g) minimum per contact.

### 7.3.3 Unitized connectors

The 40-pin I/O signal header and the 4-pin power connector may be implemented in one of two unitized connectors that provide additional pins for configuration jumpers. The dimensioning of the 40-pin I/O signal area shall be as defined in Figure 5 and the dimensioning of the 4-pin power connector area shall be as defined in Figure 12 for both unitized connectors.

The first of the unitized connectors is shown in Figure 14 with dimensions as shown in Table 20. The jumper pins, A through I, have been assigned as follows:

- E-F - CSEL
- G-H - Master
- G-H and E-F - Master with slave present
- No jumper - Slave
- A through D - Vendor specific
- I - Reserved

The second of the unitized connectors is shown in Figure 15 with dimensions as shown in Table 21. The jumper pins, A through J, have been assigned as follows:

- A-B - CSEL
- C-D - Slave
- E-F - Master
- G through J - Vendor specific

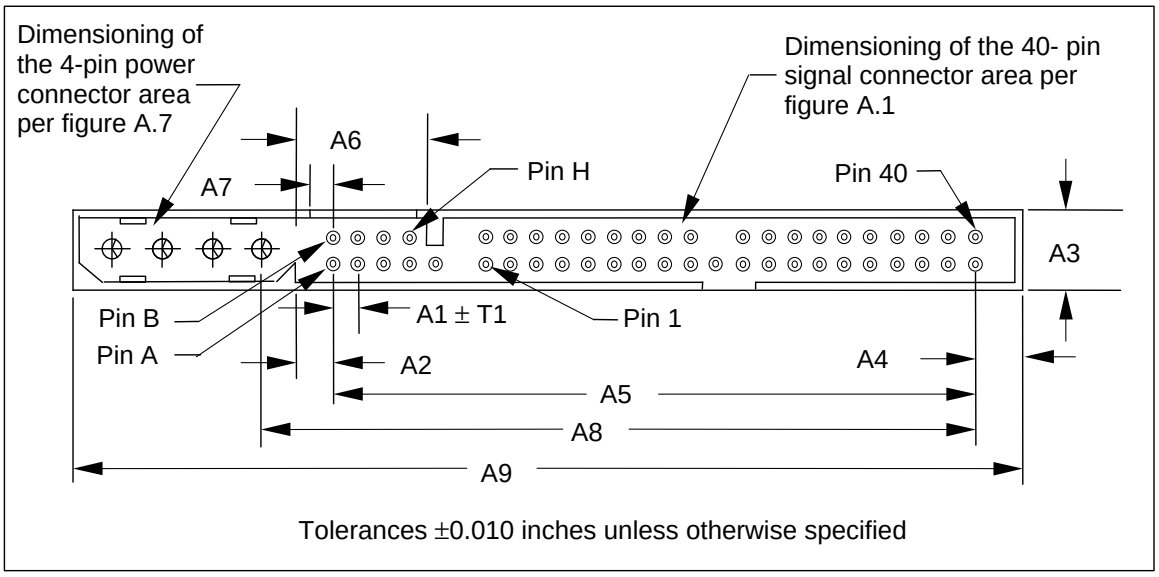



**Figure 14 – Unitized connector**

**Table 20 – Unitized connector**

| Dimension | Millimeters | Inches |
|---|---|---|
| A 1 | 2.54 | 0.100 |
| A 2 | 4.06 | 0.160 |
| A 3 | 8.40 | 0.331 |
| A 4 | 5.26 | 0.207 |
| A 5 | 63.50 | 2.500 |
| A 6 | 13.54 | 0.533 |
| A 7 | 2.54 | 0.100 |
| A 8 | 70.825 | 2.788 |
| A 9 | 95.50 | 3.760 |
| T 1 | 0.15 | 0.006 |

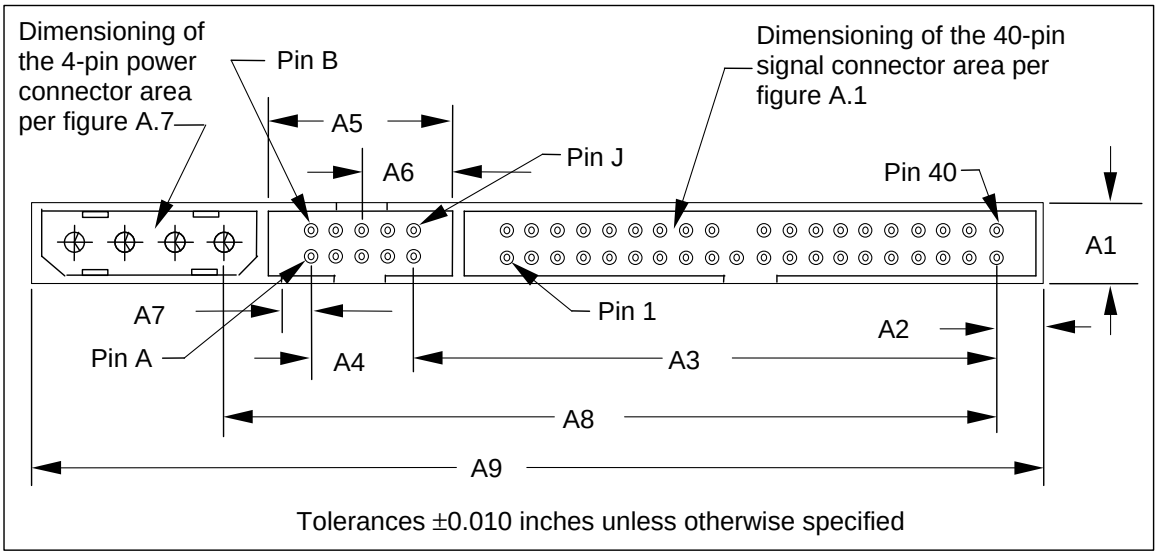



**Figure 15 – Unitized connector**

**Table 21 – Unitized connector**

| Dimension | Millimeters | Inches |
|---|---|---|
| A 1 | 8.51 | 0.335 |
| A 2 | 5.51 | 0.217 |
| A 3 | 57.15 | 2.250 |
| A 4 | 10.16 | 0.400 |
| A 5 | 17.88 | 0.704 |
| A 6 | 8.94 | 0.352 |
| A 7 | 2.54 | 0.100 |
| A 8 | 75.29 | 2.964 |
| A 9 | 100.33 | 3.950 |

### 7.3.4 50-pin 2.5 inch form factor style connector

An alternative connector is often used for 2 1/2 inch or smaller devices. This connector is shown in Figure 16 with dimensions shown in Table 22. Signal assignments are shown in Table 23. Although there are 50 pins in the plug, a 44-pin mating receptacle may be used.

Pins E, F, and 20 are keys and are removed.

Some devices may use pins A, B, C, and D for option selection via physical jumpers. If a device uses pins A, B, C, and D for device selection, when no jumper is present the device should be designated as Device 0.

When a jumper is present between pins B and D, the device should respond to the CSEL signal to determine the device number.

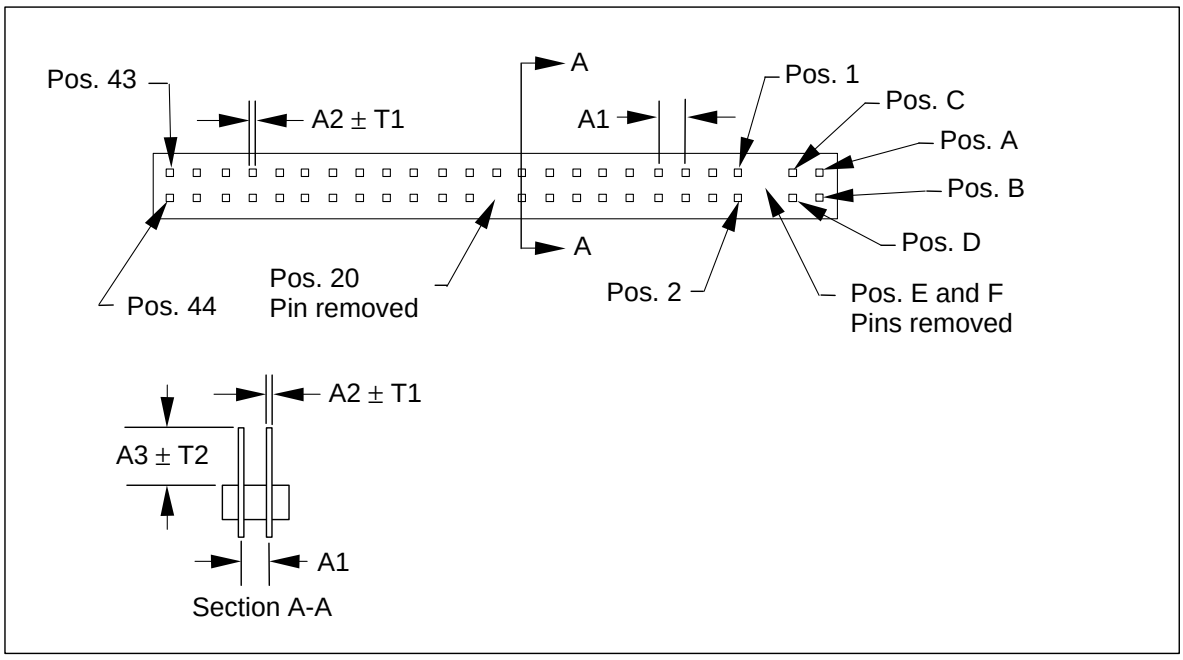



**Figure 16 – 50-pin 2.5 inch form factor style connector**

**Table 22 – 50-pin connector**

| Dimension | Millimeters | Inches |
|---|---|---|
| A 1 | 2.00 | 0.079 |
| A 2 | 0.50 | 0.020 |
| A 3 | 3.86 | 0.152 |
| T 1 | 0.05 | 0.002 |
| T 2 | 0.20 | 0.008 |

**Table 23 – Signal assignments for 50-pin 2.5 inch form factor style connector**

| Signal name | Connector contact | Conductor | | Connector contact | Signal name |
|---|---|---|---|---|---|
| Option selection pins | A | | | B | Option selection pins |
| Option selection pins | C | | | D | Option selection pins |
| (keypin) | E | | | F | (keypin) |
| RESET- | 1 | 1 | 2 | 2 | Ground |
| DD7 | 3 | 3 | 4 | 4 | DD8 |
| DD6 | 5 | 5 | 6 | 6 | DD9 |
| DD5 | 7 | 7 | 8 | 8 | DD10 |
| DD4 | 9 | 9 | 10 | 10 | DD11 |
| DD3 | 11 | 11 | 12 | 12 | DD12 |
| DD2 | 13 | 13 | 14 | 14 | DD13 |
| DD1 | 15 | 15 | 16 | 16 | DD14 |
| DD0 | 17 | 17 | 18 | 18 | DD15 |
| Ground | 19 | 19 | 20 | 20 | (keypin) |
| DMARQ | 21 | 21 | 22 | 22 | Ground |
| DIOW-:STOP | 23 | 23 | 24 | 24 | Ground |
| DIOR-:HDMARDY-:HSTROBE | 25 | 25 | 26 | 26 | Ground |
| IORDY:DDMARDY-:DSTROBE | 27 | 27 | 28 | 28 | CSEL |
| DMACK- | 29 | 29 | 30 | 30 | Ground |
| INTRQ | 31 | 31 | 32 | 32 | Obsolete (see note) |
| DA1 | 33 | 33 | 34 | 34 | PDIAG- |
| DA0 | 35 | 35 | 36 | 36 | DA2 |
| CS0- | 37 | 37 | 38 | 38 | CS1- |
| DASP- | 39 | 39 | 40 | 40 | Ground |
| +5 V (logic) | 41 | 41 | 42 | 42 | +5 V (motor) |
| Ground(return) | 43 | 43 | 44 | 44 | Reserved - no connection |
| NOTE – Pin 32 was defined as IOCS16 in ATA-2, ANSI X3.279-1996. | | | | | |

### 7.3.5 68-pin PCMCIA connector

This clause defines the pinouts used for the 68-pin alternative connector for the AT Attachment Interface. This connector is defined in the PCMCIA PC Card Standard. This clause defines a pinout alternative that allows a device to function as an AT Attachment Interface compliant device, while also allowing the device to be compliant with PC Card ATA mode defined by PCMCIA. The signal protocol allows the device to identify the host interface as being 68-pin as defined in this standard or PC Card ATA.

To simplify the implementation of dual-interface devices, the 68-pin AT Attachment Interface maintains commonality with as many PC Card ATA signals as possible, while supporting full command and signal compliance with this standard.

The 68-pin pinout shall not cause damage or loss of data if a PCMCIA card is accidentally plugged into a host slot supporting this interface. The inversion of the RESET signal between this standard and PCMCIA interfaces prevents loss of data if the device is unable to reconfigure itself to the appropriate host interface.

#### 7.3.5.1 Signals

This specification relies upon the electrical and mechanical characteristics of PCMCIA and unless otherwise noted, all signals and registers with the same names as PCMCIA signals and registers have the same meaning as defined in PCMCIA.

The PC Card ATA specification is used as a reference to identify the signal protocol used to identify the host interface protocol.

#### 7.3.5.2 Signal descriptions

Any signals not defined below shall be as described in this standard, PCMCIA, or the PC Card ATA documents.

Table 24 shows the signals and relationships such as direction, as well as providing the signal name of the PCMCIA equivalent.

**Table 24 – Signal assignments for 68-pin connector**

| Pin | Signal | Hst | Dir | Dev | PCMCIA | | Pin | Signal | Hst | Dir | Dev | PCMCIA |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Ground | x | → | x | Ground | | 35 | Ground | x | → | x | Ground |
| 2 | DD3 | x | ↔ | x | D3 | | 36 | CD1- | x | ← | x | CD1- |
| 3 | DD4 | x | ↔ | x | D4 | | 37 | DD11 | x | ↔ | x | D11 |
| 4 | DD5 | x | ↔ | x | D5 | | 38 | DD12 | x | ↔ | x | D12 |
| 5 | DD6 | x | ↔ | x | D6 | | 39 | DD13 | x | ↔ | x | D13 |
| 6 | DD7 | x | ↔ | x | D7 | | 40 | DD14 | x | ↔ | x | D14 |
| 7 | CS0- | x | → | x | CE1- | | 41 | DD15 | x | ↔ | x | D15 |
| 8 | | | → | i | A10 | | 42 | CS1- | x | → | x(1) | CE2- |
| 9 | SELATA- | x | → | x | OE- | | 43 | | | ← | i | VS1- |
| 10 | | | | | | | 44 | DIOR- | x | → | x | IORD- |
| 11 | CS1- | x | → | x(1) | A9 | | 45 | DIOW- | x | → | x | IOWR- |
| 12 | | | → | i | A8 | | 46 | | | | | |
| 13 | | | | | | | 47 | | | | | |
| 14 | | | | | | | 48 | | | | | |
| 15 | | | → | i | WE- | | 49 | | | | | |
| 16 | INTRQ | x | ← | x | READY/ IREQ- | | 50 | | | | | |
| 17 | Vcc | x | → | x | Vcc | | 51 | Vcc | x | → | x | Vcc |
| 18 | | | | | | | 52 | | | | | |
| 19 | | | | | | | 53 | | | | | |
| 20 | | | | | | | 54 | | | | | |
| 21 | | | | | | | 55 | M/S- | x | → | x(2) | |
| 22 | | | → | i | A7 | | 56 | CSEL | x | → | x(2) | |
| 23 | | | → | i | A6 | | 57 | | | ← | i | VS2- |
| 24 | | | → | i | A5 | | 58 | RESET- | x | → | x | RESET |
| 25 | | | → | i | A4 | | 59 | IORDY | o | ← | x(3) | WAIT- |
| 26 | | | → | i | A3 | | 60 | DMARQ | o | ← | x(3) | INPACK- |
| 27 | DA2 | x | → | x | A2 | | 61 | DMACK- | o | → | o | REG- |
| 28 | DA1 | x | → | x | A1 | | 62 | DASP- | x | ↔ | x | BVD2/ SPKR- |
| 29 | DA0 | x | → | x | A0 | | 63 | PDIAG- | x | ↔ | x | BVD1/ STSCHG |
| 30 | DD0 | x | ↔ | x | D0 | | 64 | DD8 | x | ↔ | x | D8 |
| 31 | DD1 | x | ↔ | x | D1 | | 65 | DD9 | x | ↔ | x | D9 |
| 32 | DD2 | x | ↔ | x | D2 | | 66 | DD10 | x | ↔ | x | D10 |
| 33 | | x | ← | x | WP/ IOIS16 | | 67 | CD2- | x | ← | x | CD2- |
| 34 | Ground | x | → | x | Ground | | 68 | Ground | x | → | x | Ground |

Key:
Dir = the direction of the signal between host and device.
x in the Hst column = this signal shall be supported by the Host.
x in the Dev column = this signal shall be supported by the device.
i in the Dev column = this signal shall be ignored by the device while in 68-pin mode.
o = this signal is Optional.
Nothing in Dev column = no connection should be made to that pin.

NOTES –
1 The device shall support only one CS1- signal pin.
2 The device shall support either M/S- or CSEL but not both.
3 The device shall hold this signal negated if it does not support the function.

#### 7.3.5.2.1 CD1- (Card Detect 1)

This signal shall be grounded by the device. CD1- and CD2- are used by the host to detect the presence of the device.

#### 7.3.5.2.2 CD2- (Card Detect 2)

This signal shall be grounded by the device. CD1- and CD2- are used by the host to detect the presence of the device.

#### 7.3.5.2.3 CS1- (Device chip select 1)

Hosts shall provide CS1- on both the pins identified in Table 24.

Devices shall recognize only one of the two pins as CS1-.

#### 7.3.5.2.4 DMACK- (DMA acknowledge)

This signal is optional for hosts and devices.

If this signal is supported by the host or the device, the function of DMARQ shall also be supported.

#### 7.3.5.2.5 DMARQ (DMA request)

This signal is optional for hosts.

If this signal is supported by the host or the device, the function of DMACK- shall also be supported.

#### 7.3.5.2.6 IORDY (I/O channel ready)

This signal is optional for hosts.

#### 7.3.5.2.7 M/S- (Master/slave)

This signal is the inverted form of CSEL. Hosts shall support both M/S- and CSEL though devices need only support one or the other.

Hosts shall assert CSEL and M/S- prior to applying VCC to the connector.

#### 7.3.5.2.8 SELATA- (Select 68-pin ATA)

This pin is used by the host to select which mode to use, PC Card ATA mode or the 68-pin mode defined in this standard. To select 68-pin ATA mode, the host shall assert SELATA- prior to applying power to the connector, and shall hold SELATA- asserted.

The device shall not re-sample SELATA- as a result of either a hardware or software reset. The device shall ignore all interface signals for 19 ms after the host supplies Vcc within the device's voltage tolerance. If SELATA- is negated following this time, the device shall either configure itself for PC Card ATA mode or not respond to further inputs from the host.

### 7.3.5.3 Removability considerations

This specification supports the removability of devices that use the protocol. As removability is a new consideration for devices, several issues need to be considered with regard to the insertion or removal of devices.

#### 7.3.5.3.1 Device recommendations

The following are recommendations to device implementors:

- CS0-, CS1-, RESET-, and SELATA- signals be negated on the device to prevent false selection during hot insertion.
- Ignore all interface signals except SELATA- until 19 ms after the host supplies $V_{CC}$ within the device's voltage tolerance. This time is necessary to de-bounce the device's power-on reset sequence. Once in the 68-pin mode as defined in this standard, if SELATA- is ever negated following the 19 ms de-bounce delay time, the device disables itself until $V_{CC}$ is removed.
- Provide a method to prevent unexpected removal of the device or media.

#### 7.3.5.3.2 Host recommendations

The following are recommendations to host implementors:

- Connector pin sequencing to protect the device by making contact to ground before any other signal in the system.
- SELATA- to be asserted at all times.
- All devices reset and reconfigured to the same base address each time a device at that address is inserted or removed.
- The removal or insertion of a device at the same address to be detected so as to prevent the corruption of a command.
- Provide a method to prevent unexpected removal of the device or media.

### 7.3.6 CompactFlash™ connector

Device compliant with the CompactFlash™ Association Specification use the connector defined in that specification.

### 7.3.7 1.8 inch 3.3V parallel connector

The connector for the 1.8 inch 3.3V parallel form factor device is defined in Figure 17 with dimensions defines in Table 25. Pin assignments are defined in Table 27. The host connector for the 1.8 inch 3.3V parallel form factor device is defined in Figure 18 with dimensions defines in Table 26. Pin assignments are defined in Table 27.

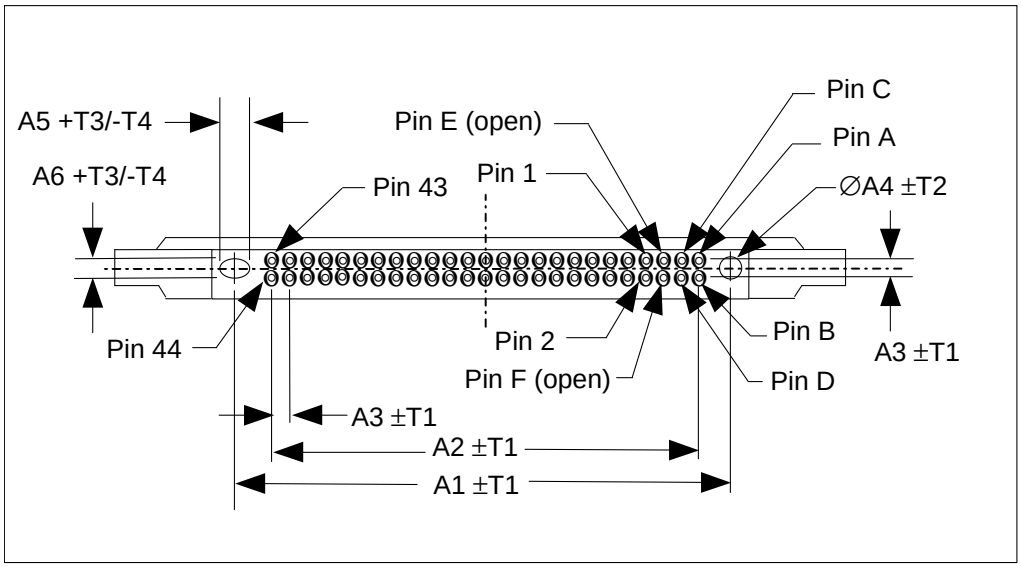



**Figure 17 – 1.8 inch 3.3V parallel connector**

**Table 25 – 1.8 inch 3.3V parallel connector**

| Dimension | Value (inches) |
|---|---|
| A1 | 1.445 |
| A2 | 1.200 |
| A3 | 0.050 |
| A4 | 0.083 |
| A5 | 0.098 |
| A6 | 0.059 |
| T1 | 0.006 |
| T2 | 0.002 |
| T3 | 0.004 |
| T4 | 0.000 |

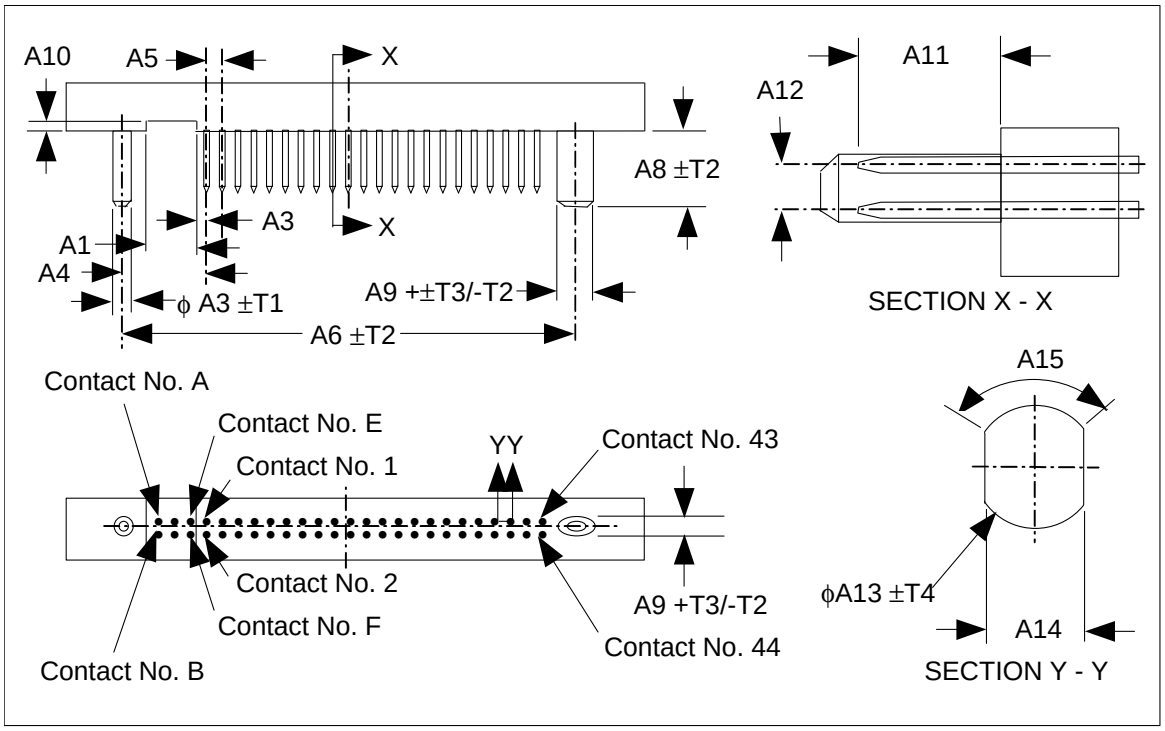



**Figure 18 – 1.8 inch 3.3 V parallel host connector**

**Table 26 – 1.8 inch 3.3 V parallel host connector**

| Dimension | Value (inches) |
|---|---|
| A1 | 0.170 |
| A2 | 0.075 |
| A3 | 0.025 |
| A4 | 0.272 |
| A5 | 0.050 |
| A6 | 1.445 |
| A7 | 0.091 |
| A8 | 0.177 |
| A9 | 0.055 |
| A10 | 0.024 |
| A11 | 0.138 |
| A12 | 0.050 |
| A13 | 0.017 |
| A14 | 0.017 max |
| A15 | 110 ° min |
| T1 | 0.002 |
| T2 | 0.004 |
| T3 | 0.000 |
| T4 | 0.0008 |

**Table 27 – Pin assignments for the 1.8 inch 3.3V parallel connector**

| Pin | Signal | Pin | Signal |
|---|---|---|---|
| 1 | RESET- | 2 | Ground |
| 3 | DD7 | 4 | DD8 |
| 5 | DD6 | 6 | DD9 |
| 7 | DD5 | 8 | DD10 |
| 9 | DD4 | 10 | DD11 |
| 11 | DD3 | 12 | DD12 |
| 13 | DD2 | 14 | DD13 |
| 15 | DD1 | 16 | DD14 |
| 17 | DD0 | 18 | DD15 |
| 19 | Ground | 20 | No connection |
| 21 | DMARQ | 22 | Ground |
| 23 | DIOW-:STOP | 24 | Ground |
| 25 | DIOR-:HDMARDY-:HSTROBE | 26 | Ground |
| 27 | IORDY:DDMARDY-:DSTROBE | 28 | CSEL |
| 29 | DMACK- | 30 | Ground |
| 31 | INTRQ | 32 | IOCS16- |
| 33 | DA1 | 34 | PDIAG-:CBLID- |
| 35 | DA0 | 36 | DA2 |
| 37 | CS0- | 38 | CS1- |
| 39 | DASP- | 40 | Ground |
| 41 | +3.3 V | 42 | -3.3 V (motor) |
| 43 | Ground | 44 | Reserved |

## 7.4 Physical form factors

### 7.4.1 3.5” form factor

The 3.5” form factor is shown in Figure 19 with dimensions shown in Table 28. Physical dimensions shall be measured at 20 ± 2 degrees C. The device shall not be exposed to any conditions (transit temperatures, shock, etc.) beyond the manufacturer’s specified limits before measurement.

The position of four mounting holes on the bottom of the device and two mounting holes on each side of the device are specitfied.

The Device PCB may extend beyond the HDA. If it does, there is a space at the end of the HDA underneath or above the PCB overhang. Such dead space shall not be encroached upon by the host system.

Except for attachment points, 0.75 mm clearance around the device is recommended for cooling airflow.

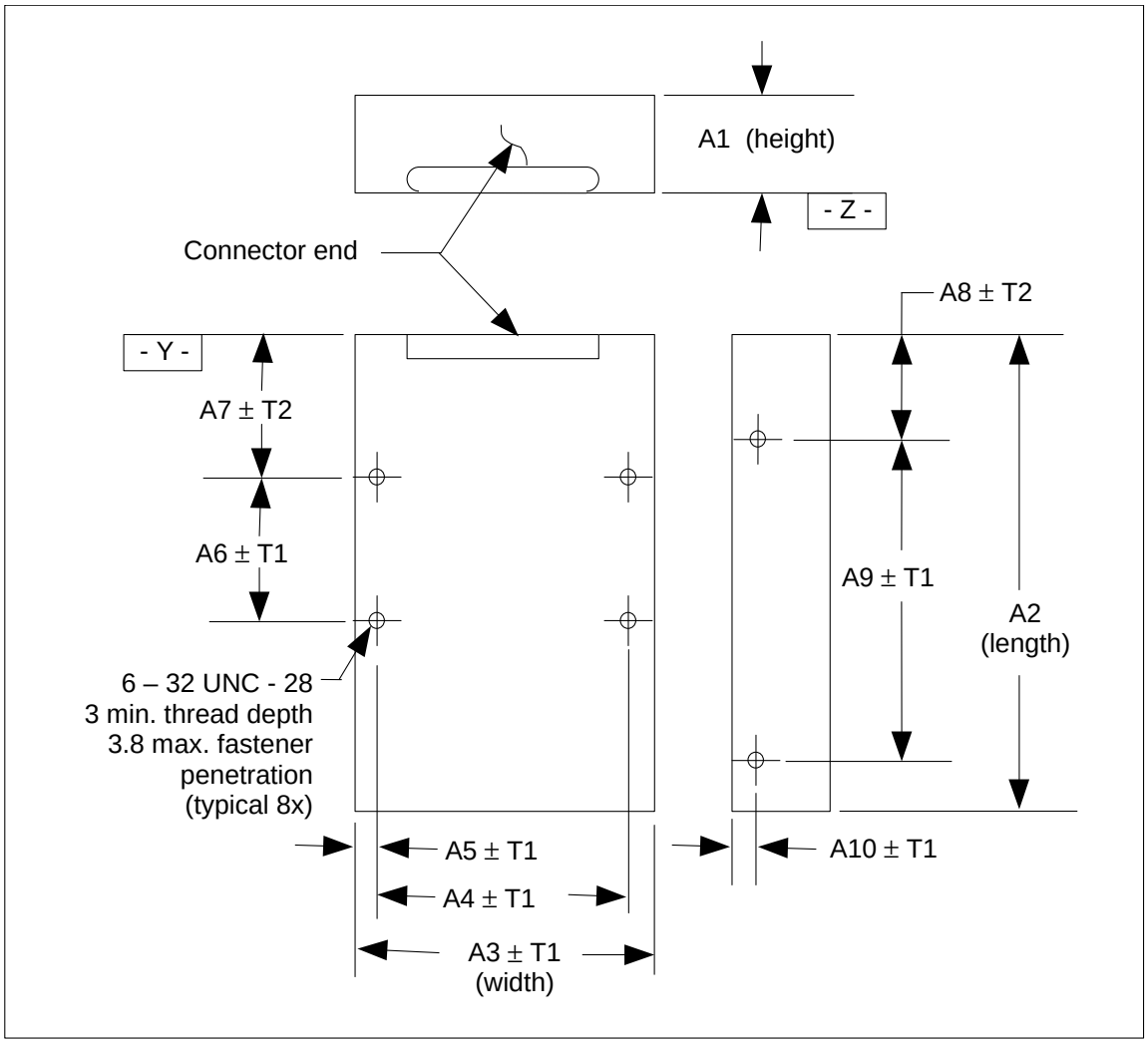



**Figure 19 – 3.5” form factor**

**Table 28 – 3.5” form factor**

| Dimension | Value (inches) |
|---|---|
| A1 | 1.028 max |
| A1 | 1.654 max |
| A2 | 5.787 max |
| A3 | 4.000 |
| A4 | 3.750 |
| A5 | 0.125 |
| A6 | 1.750 |
| A7 | 1.625 |
| A8 | 1.122 |
| A9 | 4.000 |
| A10 | 0.250 |
| T1 | 0.010 |
| T2 | 0.020 |

#### 7.4.1.1 Connector location for 3.5” form factor

A 3.5” form factor device may may use the 40-pin signal connector (see 7.3.1) and the 4-pin power connector (see 7.3.2) or one of the Unitized connectors (see 7.3.3). The location of connectors on the 3.5” form factor is not specified.

### 7.4.2 2.5” form factor

The 2.5” form factor is shown in Figure 20 with dimensions shown in Table 29. Physical dimensions shall be measured at 20 ± 2 degrees C. The device shall not be exposed to any conditions (transit temperatures, shock, etc.) beyond the manufacturer’s specified limits before measurement.

Sixteen mounting holes are defined in Figure 20. Only eight of these mounting holes need be implemented as shown in Figure 21.

The Device PCB may extend beyond the HDA. If it does, there is a space at the end of the HDA underneath or above the PCB overhang. Such dead space shall not be encroached upon by the host system.

Except for attachment points, 0.75 mm clearance around the device is recommended for cooling airflow.

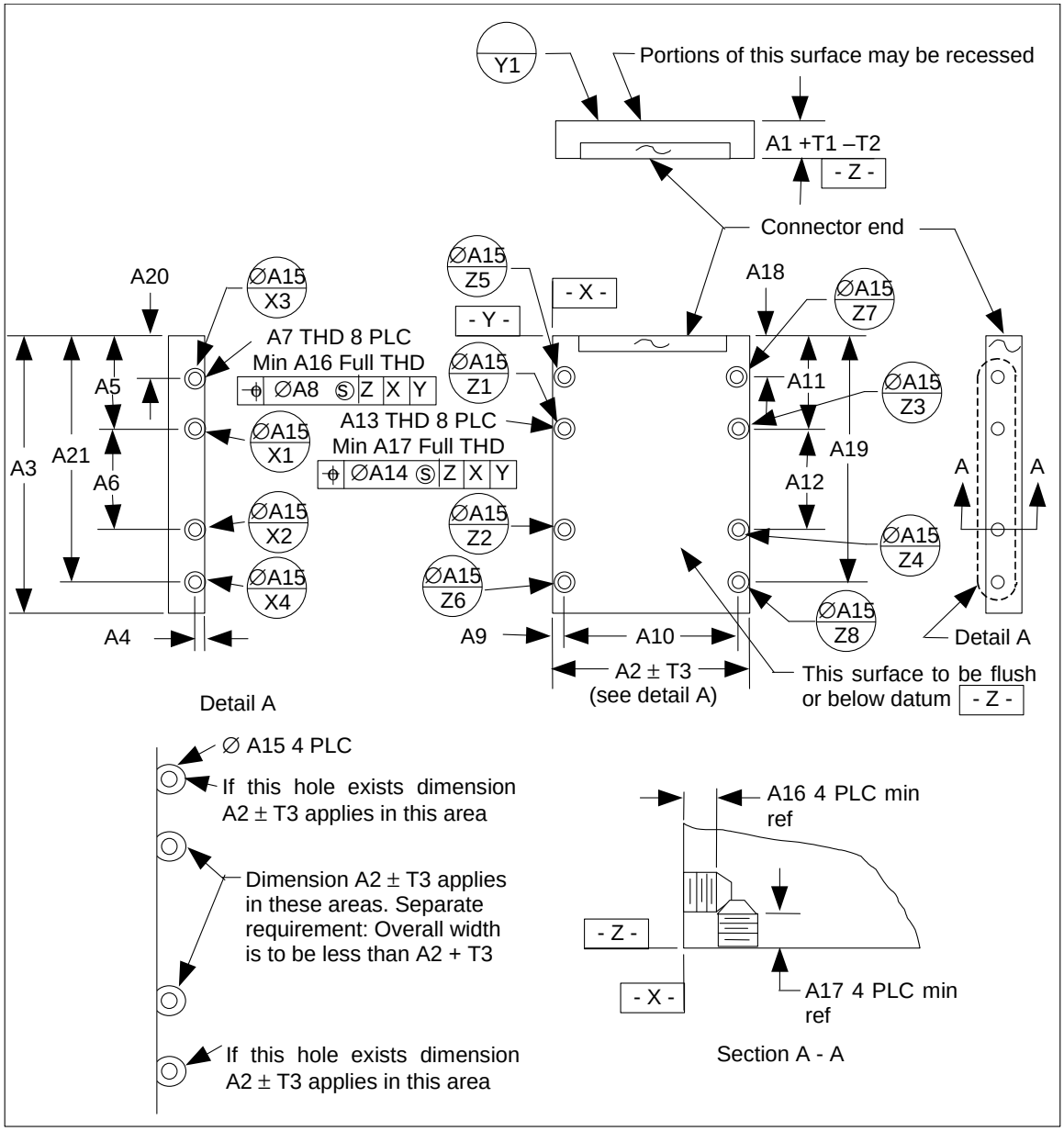



**Figure 20 – 2.5” form factor**

**Table 29 – 2.5“ form factor**

| Dimension | Value (inches) |
|---|---|
| A1 | 0.750 |
| A1 | 0.669 |
| A1 | 0.591 |
| A1 | 0.500 |
| A1 | 0.413 |
| A1 | 0.374 |
| A1 | 0.333 |
| A1 | 0.276 |
| A2 | 2.750 |
| A3 | 4.010 max |
| A4 | 0.118 |
| A5 | 1.375 |
| A6 | 1.500 |
| A7 | n/a |
| A8 | 0.020 |
| A9 | 0.160 |
| A10 | 2.430 |
| A11 | 1.375 |
| A12 | 1.500 |
| A13 | n/a |
| A14 | 0.020 |
| A15 | 0.315 |
| A16 | 0.118 min |
| A17 | 0.098 min |
| A18 | 0.551 |
| A19 | 3.567 |
| A20 | 0.551 |
| A21 | 3.567 |
| T1 | 0.000 |
| T2 | 0.020 |
| T3 | 0.010 |

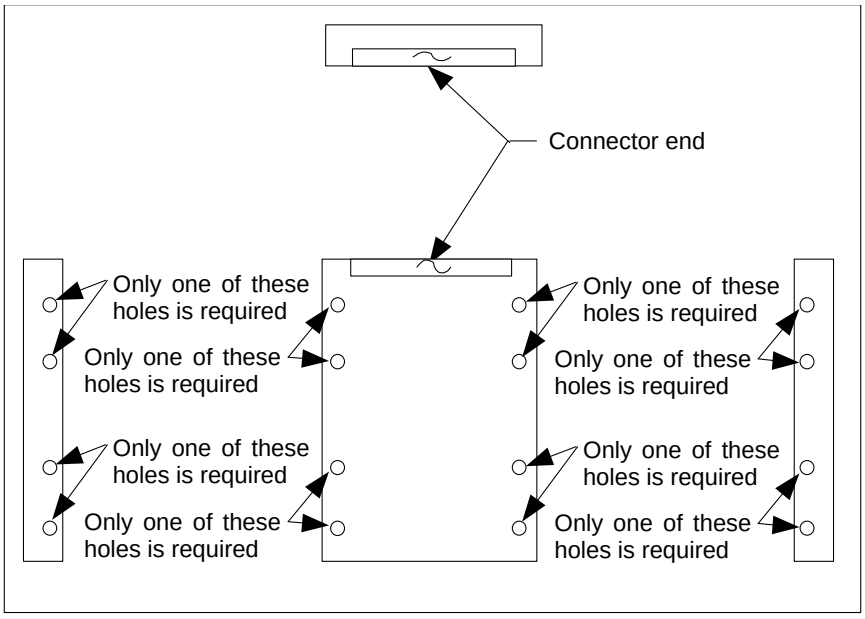



**Figure 21 –2.5” form factor mounting holes**

#### 7.4.2.1 Connector location for 2.5” form factor

A 2.5” form factor device shall use the 50-pin connector (see 7.3.4). The location of the connector on the 2.5” form factor is defined in Figure 22 with dimensions shown in Table 30.

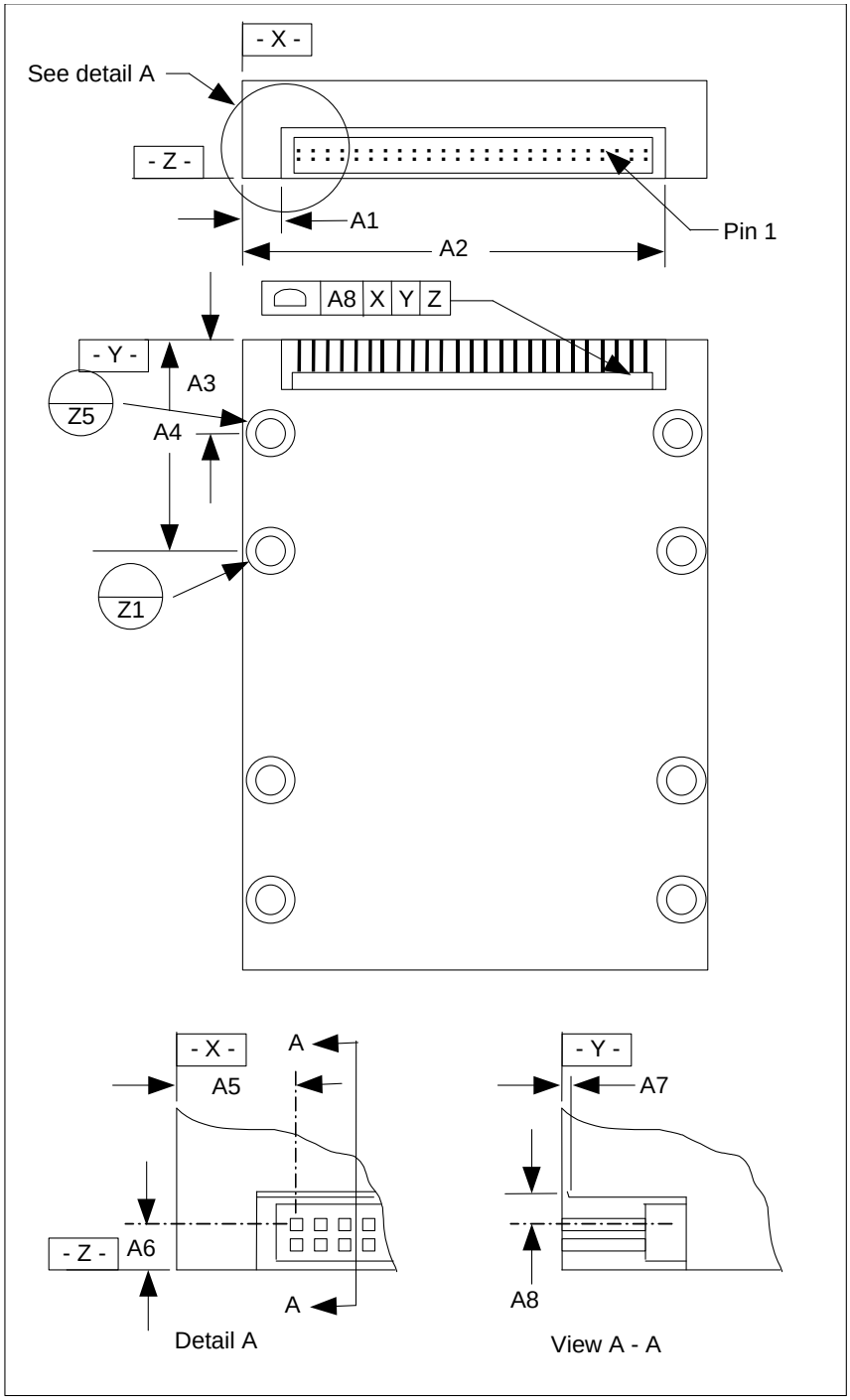



**Figure 22 – 2.5” form factor connector location**

**Table 30 – 2.5 inch form factor connector location**

| Dimension | Value (inches) |
|---|---|
| A1 | 0.315 max |
| A2 | 2.370min |
| A3 | 0.403 |
| A4 | 1.227 |
| A5 | 0.399 |
| A6 | 0.157 |
| A7 | 0.010 min |
| A8 | 0.049min |

### 7.4.3 1.8” PCMCIA form factor

The 1.8” PCMCIA form factor is defined in the PC Card Standard (see **Error! Reference source not found.**).

#### 7.4.3.1 Connector location for 1.8” PCMCIA form factor

The connector location for the 1.8” PCMCIA form factor is defined in the PC Card Standard (see **Error! Reference source not found.**).

### 7.4.4 1.8” 5V parallel form factor

The 1.8” 5V parallel form factor is shown in Figure 23 with dimensions shown in Table 31. Physical dimensions shall be measured at 20 ± 2 degrees C. The device shall not be exposed to any conditions (transit temperatures, shock, etc.) beyond the manufacturer’s specified limits before measurement.

Eight mounting holes are defined in Figure 23. All of these mounting holes shall be implemented.

The Device PCB may extend beyond the HDA. If it does, there is a space at the end of the HDA underneath or above the PCB overhang. Such dead space shall not be encroached upon by the host system.

Except for attachment points, 0.75 mm clearance around the device is recommended for cooling airflow.

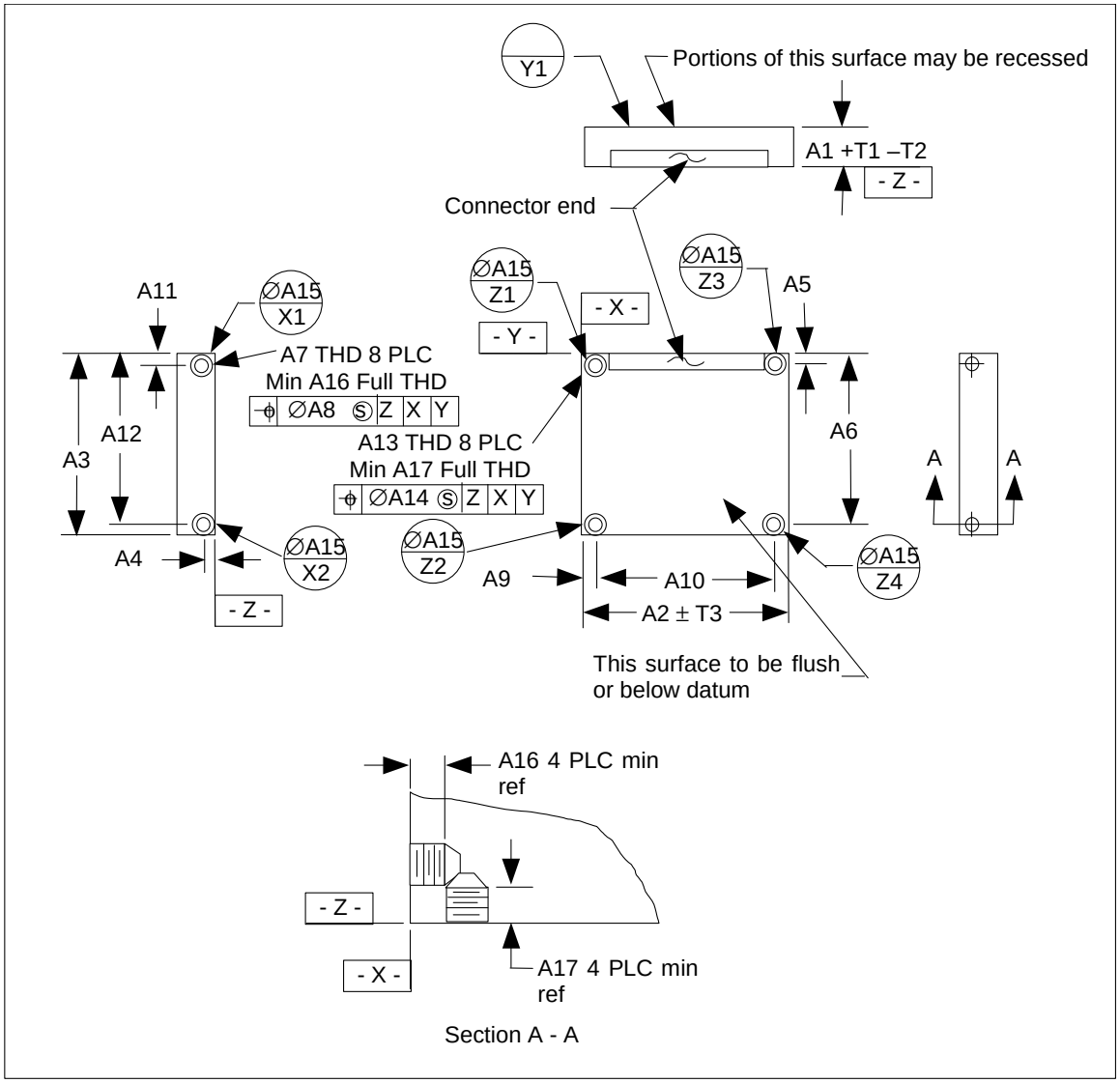



**Figure 23 – 1.8 inch 5V parallel form factor**

**Table 31 – 1.8 inch 5V parallel form factor**

| Dimension | Value (inches) |
|---|---|
| A1 | 0.374 |
| A1 | 0.276 |
| A2 | 2.700 |
| A3 | 2.362 |
| A4 | 0.111 |
| A5 | 0.116 |
| A6 | 2.246 |
| A7 | 0.000 |
| A8 | 0.020 |
| A9 | 0.126 |
| A10 | 2.498 |
| A11 | 0.116 |
| A12 | 2.246 |
| A13 | 0.000 |
| A14 | 0.020 |
| A15 | 0.157 |
| A16 | 0.110 |
| A17 | 0.110 |
| T1 | 0.000 |
| T2 | 0.020 |
| T3 | 0.010 |

#### 7.4.4.1 Connector location for 1.8” 5V parallel form factor

A 1.8” 5V parallel form factor device shall use the 50-pin connector (see 7.3.4). The location of the connector on the device is defined in Figure 24 with dimensions shown in Table 32.

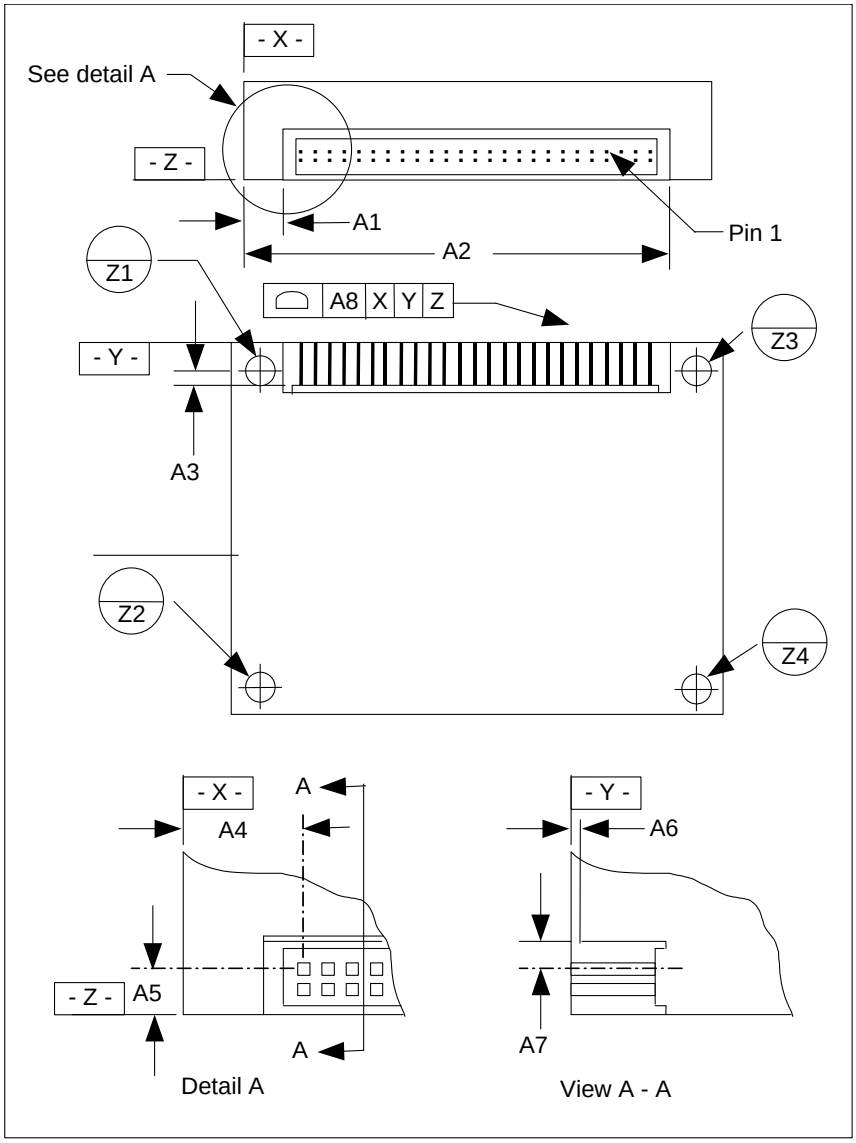



**Figure 24 – 1.8 inch 5V parallel form factor connector location**

**Table 32 – 1.8 inch 5V parallel form factor connector location**

| Dimension | Value (inches) |
|---|---|
| A1 | 0.315 max |
| A2 | 2.370 min |
| A3 | 0.036 |
| A4 | 0.399 |
| A5 | 0.157 |
| A6 | 0.010 min |
| A7 | 0.049 min |

### 7.4.5 1.8” 3.3V parallel form factor

The 1.8” 3.3 V parallel form factor is shown in Figure 25 with dimensions shown in Table 33. Physical dimensions shall be measured at 20 ± 2 degrees C. The device shall not be exposed to any conditions (transit temperatures, shock, etc.) beyond the manufacturer’s specified limits before measurement.

The mounting area is defined in Figure 25. The device shall be guided or fixed by the interconnect or mounting area.

Except for attachment points, 0.75 mm clearance around the device is recommended for cooling airflow.

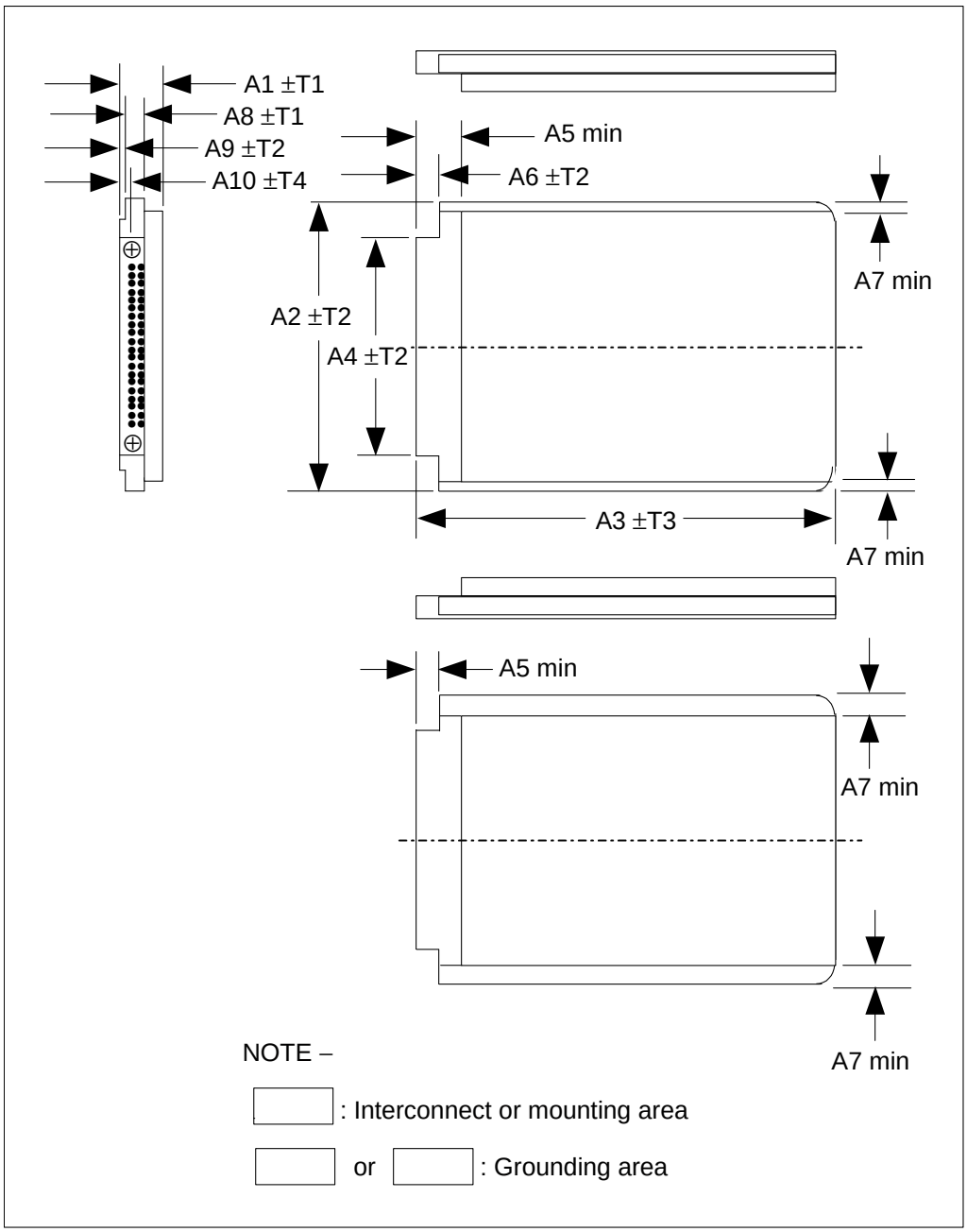



**Figure 25 – 1.8 inch 3.3V parallel form factor**

**Table 33 – 1.8 inch 3.3V parallel form factor**

| Dimension | Value (inches) |
|---|---|
| A1 | 0.197 |
| A1 | 0.315 |
| A2 | 2.126 |
| A3 | 3.090 |
| A4 | 1.575 |
| A5 | 0.256 |
| A6 | 0.138 |
| A7 | 0.059 |
| A8 | 0.130 |
| A9 | 0.033 |
| A10 | 0.065 |
| T1 | 0.006 |
| T2 | 0.008 |
| T3 | 0.012 |
| T4 | 0.004 |

#### 7.4.5.1 Connector location for 1.8” 3.3V parallel form factor

The connector location for the 1.8 inch 3.3V parallel connector is described in 7.3.7.

### 7.4.6 5.25” form factor

#### 7.4.6.1 5.25” HDD form factor

The 5.25” HDD form factor is shown in Figure 26 with dimensions shown in Table 34. Physical dimensions shall be measured at 20 ± 2 degrees C. The device shall not be exposed to any conditions (transit temperatures, shock, etc.) beyond the manufacturer’s specified limits before measurement.

The position of four mounting holes on the bottom of the device and four mounting holes on each side of the device are specitfied.

Except for attachment points, 0.75 mm clearance around the device is recommended for cooling airflow.

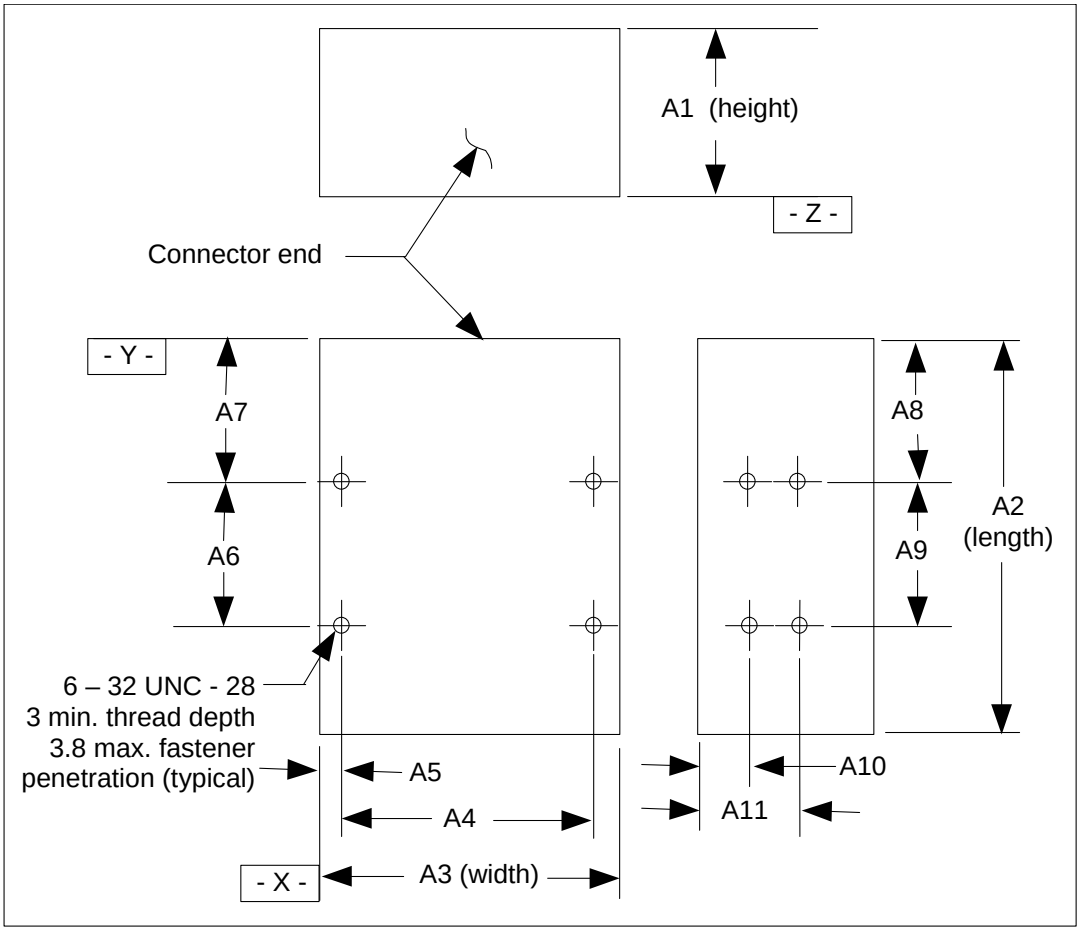



**Figure 26 – 5.25 inch HDD form factor**

**Table 34 – 5.25 inch HDD form factor**

| Dimension | Value (inches) |
|---|---|
| A1 | 3.250 max |
| A2 | 8.060 |
| A3 | 5.750 |
| A4 | 5.500 |
| A5 | 0.120 |
| A6 | 3.120 |
| A7 | 3.161 |
| A8 | 3.157 |
| A9 | 3.120 |
| A10 | 0.390 |
| A11 | 0.860 |

#### 7.4.6.1.1 5.25 inch HDD form factor connector location

A 5.25” HDD form factor device may may use the 40-pin signal connector (see 7.3.1) and the 4-pin power connector (see 7.3.2) or one of the Unitized connectors (see 7.3.3). The location of connectors on the 5.25” HDD form factor is not specified.

#### 7.4.6.2 5.25 inch CD-ROM form factor

The 5.25” CD-ROM form factor is shown in Figure 27 with dimensions shown in Table 35. Physical dimensions shall be measured at 20 ± 2 degrees C. The device shall not be exposed to any conditions (transit temperatures, shock, etc.) beyond the manufacturer’s specified limits before measurement.

The position of four mounting holes on the bottom of the device and four mounting holes on each side of the device are specitfied.

Except for attachment points, 0.75 mm clearance around the device is recommended for cooling airflow.

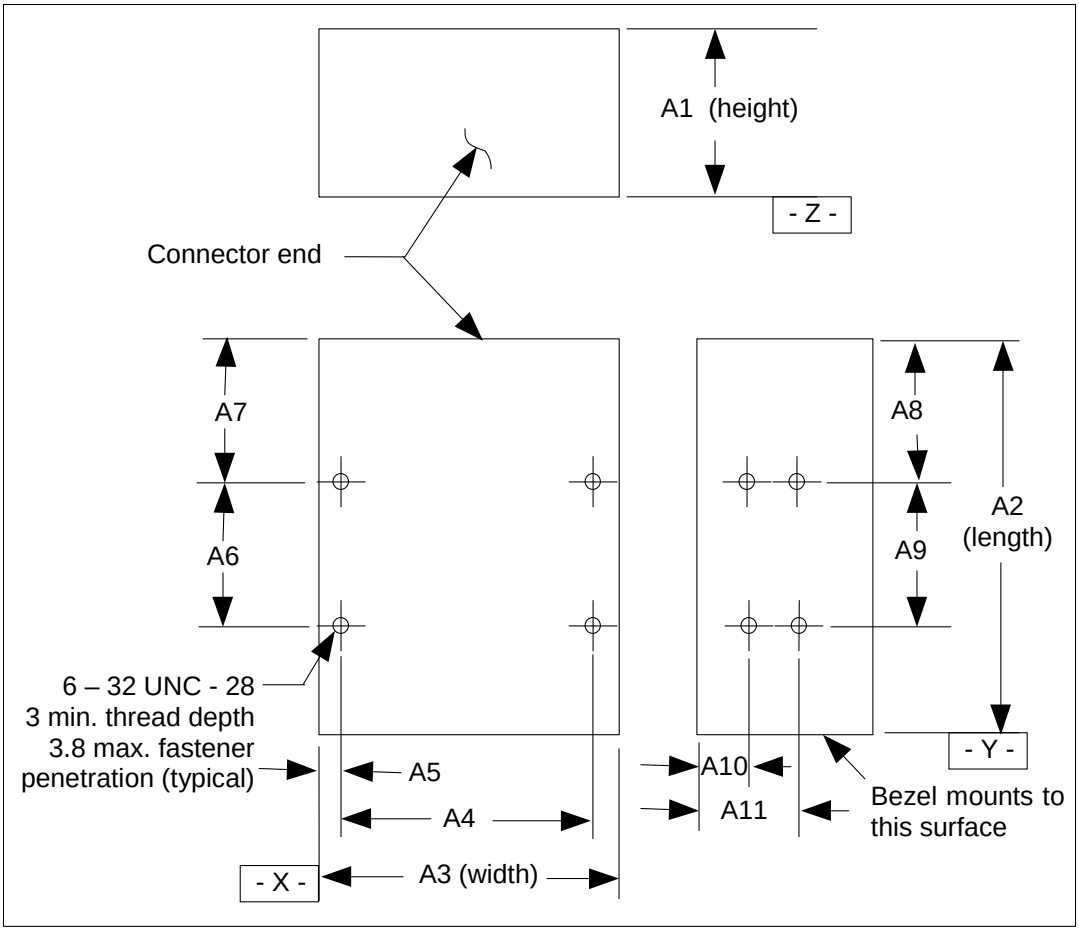



**Figure 27 – 5.25 inch CD-ROM form factor**

**Table 35 – 5.25 inch CD-ROM form factor**

| Dimension | Value (inches) |
|---|---|
| A1 | 3.250 max |
| A1 | 1.634 max |
| A1 | 1.020 max |
| A2 | 8.110 |
| A3 | 5.748 |
| A4 | 5.500 |
| A5 | 0.124 |
| A6 | 3.118 |
| A7 | 2.063 |
| A8 | 2.063 |
| A9 | 3.118 |
| A10 | 0.394 |
| A11 | 0.858 |

#### 7.4.6.2.1 5.25 inch CD-ROM form factor connector location

A 5.25” CD-ROM form factor device shall use the 40-pin signal connector (see 7.3.1) and the 4-pin power connector (see 7.3.2). The location of connectors of these connectors as well as the additional audio connectors is shown in **Figure 28**.

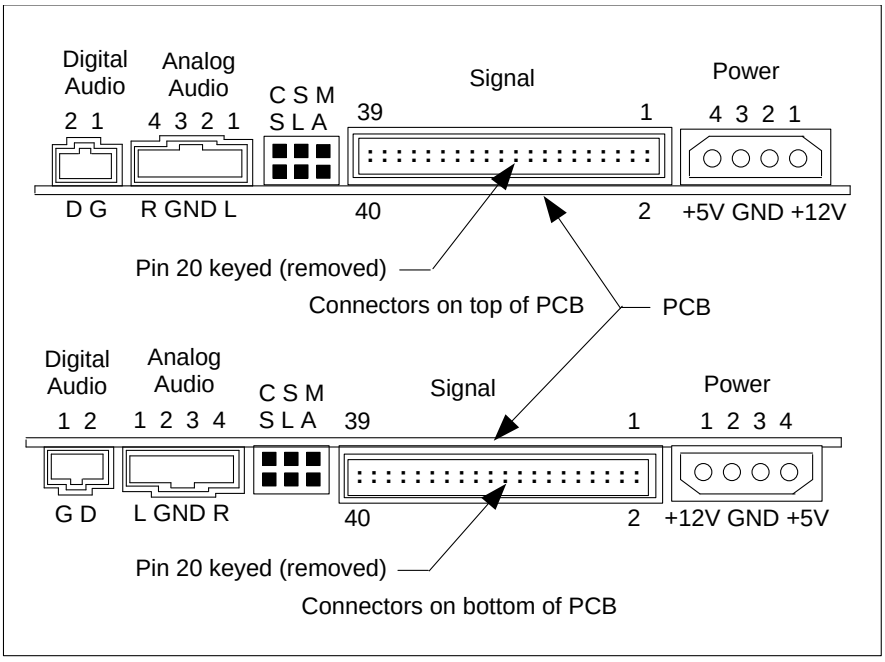



**Figure 28 – 5.25 inch CD-ROM connector location**

# 8 Parallel interface signal assignments and descriptions

## 8.1 Signal summary

The physical interface consists of receivers and drivers communicating through a set of conductors using an asynchronous interface protocol. Table 36 defines the signal names. For connector descriptions see 7.3. For driver and termination definition see 7.2.2. For signal protocol and timing see clause 11 and clause 12.

**Table 36 – Interface signal name assignments**

| Description | Host | Dir | Dev | Acronym |
|---|---|---|---|---|
| Cable select | | (see note) | | CSEL |
| Chip select 0 | | | → | CS0- |
| Chip select 1 | | | → | CS1- |
| Data bus bit 0 | | ↔ | | DD0 |
| Data bus bit 1 | | ↔ | | DD1 |
| Data bus bit 2 | | ↔ | | DD2 |
| Data bus bit 3 | | ↔ | | DD3 |
| Data bus bit 4 | | ↔ | | DD4 |
| Data bus bit 5 | | ↔ | | DD5 |
| Data bus bit 6 | | ↔ | | DD6 |
| Data bus bit 7 | | ↔ | | DD7 |
| Data bus bit 8 | | ↔ | | DD8 |
| Data bus bit 9 | | ↔ | | DD9 |
| Data bus bit 10 | | ↔ | | DD10 |
| Data bus bit 11 | | ↔ | | DD11 |
| Data bus bit 12 | | ↔ | | DD12 |
| Data bus bit 13 | | ↔ | | DD13 |
| Data bus bit 14 | | ↔ | | DD14 |
| Data bus bit 15 | | ↔ | | DD15 |
| Device active or slave (Device 1) present | | (see note) | | DASP- |
| Device address bit 0 | | | → | DA0 |
| Device address bit 1 | | | → | DA1 |
| Device address bit 2 | | | → | DA2 |
| DMA acknowledge | | | → | DMACK- |
| DMA request | ← | | | DMARQ |
| Interrupt request | ← | | | INTRQ |
| I/O read | | | → | DIOR- |
| DMA ready during Ultra DMA data-in bursts | | | → | HDMARDY- |
| Data strobe during Ultra DMA data-out bursts | | | → | HSTROBE |
| I/O ready | ← | | | IORDY |
| DMA ready during Ultra DMA data-out bursts | ← | | | DDMARDY- |
| Data strobe during Ultra DMA data-in bursts | ← | | | DSTROBE |
| I/O write | | | → | DIOW- |
| Stop during Ultra DMA data bursts | | | → | STOP |
| Passed diagnostics | | (see note) | | PDIAG- |
| Cable assembly type identifier | | (see note) | | CBLID- |
| Reset | | | → | RESET- |
| NOTE – See signal descriptions and annex A for information on source of these signals | | | | |

## 8.2 Signal descriptions

### 8.2.1 CS(1:0)- (Chip select)

These are the chip select signals from the host used to select the Command Block or Control Block registers (see 10). When DMACK- is asserted, CS0- and CS1- shall be negated and transfers shall be 16 bits wide.

### 8.2.2 DA(2:0) (Device address)

This is the 3-bit binary coded address asserted by the host to access a register or data port in the device (see 10.

### 8.2.3 DASP- (Device active, device 1 present)

During the reset protocol, DASP- shall be asserted by Device 1 to indicate that the device is present. At all other times, DASP- may be asserted by the selected active device.

### 8.2.4 DD(15:0) (Device data)

This is an 8- or 16-bit bi-directional data interface between the host and the device. DD(7:0) are used for 8-bit register transfers. Data transfers are 16-bits wide except for CFA devices that implement 8-bit data transfers.

### 8.2.5 DIOR-:HDMARDY-:HSTROBE (Device I/O read:Ultra DMA ready:Ultra DMA data strobe)

DIOR- is the strobe signal used by the host to read device registers or the Data port. Data is tranferred on the negation of this signal.

HDMARDY- is a flow control signal for Ultra DMA data-in bursts. This signal is asserted by the host to indicate to the device that the host is ready to receive Ultra DMA data-in bursts. The host may negate HDMARDY- to pause an Ultra DMA data-in burst.

HSTROBE is the data-out strobe signal from the host for an Ultra DMA data-out burst. Both the rising and falling edge of HSTROBE latch the data from DD(15:0) into the device. The host may stop generating HSTROBE edges to pause an Ultra DMA data-out burst.

### 8.2.6 DIOW-:STOP (Device I/O write:Stop Ultra DMA burst)

DIOW- is the strobe signal used by the host to write device registers or the Data port. Data is tranferred on the negation of this signal.

DIOW- shall be negated by the host prior to initiation of an Ultra DMA burst. STOP shall be negated by the host before data is transferred in an Ultra DMA burst. Assertion of STOP by the host during an Ultra DMA burst signals the termination of the Ultra DMA burst.

### 8.2.7 DMACK- (DMA acknowledge)

This signal shall be used by the host in response to DMARQ to initiate DMA transfers. For Multiword DMA transfers, the DMARQ/DMACK- handshake is used to provide flow control during the transfer. For Ultra DMA, the DMARQ/DMACK- handshake is used to indicate when the function of interface signals changes.

When DMACK- is asserted, CS0- and CS1- shall not be asserted and transfers shall be 16 bits wide.

### 8.2.8 DMARQ (DMA request)

This signal, used for DMA data transfers between host and device, shall be asserted by the device when the device is ready to transfer data to or from the host. For Mulitword DMA transfers, the direction of data transfer is controlled by DIOR- and DIOW-. This signal is used in a handshake manner with DMACK-, i.e., the device shall wait until the host asserts DMACK- before negating DMARQ, and re-asserting DMARQ if

there is more data to transfer. For Multiword DMA transfers, the DMARQ/DMACK- handshake is used to provide flow control during the transfer. For Ultra DMA, the DMARQ/DMACK- handshake is used to indicate when the function of interface signals changes.

This signal shall be released when the device is not selected.

See 9.2 and 9.3.

### 8.2.9 INTRQ (Device interrupt)

This signal is used by the selected device to interrupt the host system when interrupt pending is set. When the nIEN bit is cleared to zero and the device is selected, INTRQ shall be enabled through a tri-state buffer. When the nIEN bit is set to one or the device is not selected, the INTRQ signal shall be released.

When asserted, this signal shall be negated by the device within 400 ns of the negation of DIOR- that reads the Status register to clear interrupt pending. When asserted, this signal shall be negated by the device within 400 ns of the negation of DIOW- that writes the Command register to clear interrupt pending.

When the device is selected by writing to the Device register while interrupt pending is set, INTRQ shall be asserted within 400 ns of the negation of DIOW- that writes the Device register. When the device is deselected by writing to the Device register while interrupt pending is set, INTRQ shall be released within 400 ns of the negation of DIOW- that writes the Device register.

For devices implementing the Overlapped feature set, if INTRQ assertion is being disabled using nIEN at the same instant that the device asserts INTRQ, the minimum pulse width shall be at least 40 ns.

This signal shall be released when the device is not selected.

### 8.2.10 IORDY:DDMARDY-:DSTROBE (I/O channel ready:Ultra DMA ready:Ultra DMA data strobe)

IORDY is negated to extend the host transfer cycle of any host register access (read or write) when the device is not ready to respond to a data transfer request. If the device requires that the host transfer cycle time be extended for PIO modes 3 and above, the device shall use IORDY. Hosts that use PIO modes 3 and above shall support IORDY.

DDMARDY- is a flow control signal for Ultra DMA data-out bursts. This signal is asserted by the device to indicate to the host that the device is ready to receive Ultra DMA data-out bursts. The device may negate DDMARDY- to pause an Ultra DMA data-out burst.

DSTROBE is the data-in strobe signal from the device for an Ultra DMA data-in burst. Both the rising and falling edge of DSTROBE latch the data from DD(15:0) into the host. The device may stop generating DSTROBE edges to pause an Ultra DMA data-in burst.

This signal shall be released when the device is not selected.

### 8.2.11 PDIAG-:CBLID- (Passed diagnostics:Cable assembly type identifier)

PDIAG- shall be asserted by Device 1 to indicate to Device 0 that Device 1 has completed diagnostics (see clause 11).

The host may sample CBLID- after a power-on or hardware reset in order to detect the presence of an 80-conductor cable assembly by performing the following steps:

a) Wait until the power-on or hardware reset protocol is complete for all devices on the cable; remember which devices are present for the last step.
b) If Device 1 is not present, go to step d.
c) Issue IDENTIFY DEVICE or IDENTIFY PACKET DEVICE to Device 1. From the information returned, save word 80 and word 93 for the last step.

NOTE – Word 80 bit 3 indicates compliance with ATA-3 or subsequent standards and word 93 bits (15:13) indicate support of and results from sampling CBLID- at the device.

d) Issue IDENTIFY DEVICE or IDENTIFY PACKET DEVICE to Device 0. From the information returned, save Word 93 for the last step.

NOTE – Word 93 bits (15:13) indicate support of and results from sampling CBLID- at the device.

e) Detect the state of the CBLID- signal at the host connector and save the result for the last step.

NOTE – Any device compliant with ATA-3 or subsequent standards releases PDIAG- no later than after the first command following a power-on or hardware reset sequence and will not interfere with host detection of CBLID- in this step. Some devices claiming compliance with ATA-3 or subsequent standards are known to continue to assert CBLID-:PDIAG- which sometimes causes a 40-conductor cable assembly to be detected as an 80-conductor cable assembly.

f) Look up the output in Table 37 based on the inputs saved from steps a, c, d, and e.

**Table 37 - Cable type identification**

| Inputs | | | | Output |
|---|---|---|---|---|
| **Sensed CBLID-** | **Device 1 Word 80 bit 3** | **Device 1 Word 93 bits (15:13)** | **Device 0 Word 93 bits (15:13)** | **Cable conductors** |
| High | X | XXX | XXX | 40 |
| Low | Device absent | Device absent | 00X or 1XX | 80 |
| Low | Device absent | Device absent | 010 | Note 2 |
| Low | Device absent | Device absent | 011 | 80 |
| Low | 0 | 00X or 1XX | Device absent | Note 1 |
| Low | 0 | 00X or 1XX | 00X or 1XX | Note 1 |
| Low | 0 | 00X or 1XX | 010 | Note 2 |
| Low | 0 | 00X or 1XX | 011 | 80 |
| Low | 0 | 010 | Device absent | Note 2 |
| Low | 0 | 010 | XXX | Note 2 |
| Low | 0 | 011 | Device absent | 80 |
| Low | 0 | 011 | 00X or 1XX | 80 |
| Low | 0 | 011 | 010 | Note 2 |
| Low | 0 | 011 | 011 | 80 |
| Low | 1 | 00X or 1XX | Device absent | 80 |
| Low | 1 | 00X or 1XX | 00X or 1XX | 80 |
| Low | 1 | 00X or 1XX | 010 | Note 2 |
| Low | 1 | 00X or 1XX | 011 | 80 |
| Low | 1 | 010 | Device absent | Note 2 |
| Low | 1 | 010 | XXX | Note 2 |
| Low | 1 | 011 | Device absent | 80 |
| Low | 1 | 011 | 00X or 1XX | 80 |
| Low | 1 | 011 | 010 | Note 2 |
| Low | 1 | 011 | 011 | 80 |

NOTES –

1 Host cannot determine cable type due to insufficient information. For these cases, host should not use Ultra DMA modes higher than mode 2 without using other means to confirm presence of 80-conductor cable.

2 Host cannot determine cable type due to conflicting information. For these cases, host should not use Ultra DMA modes higher than mode 2 without using other means to confirm presence of 80-conductor cable.

3 X represents a don't-care input.

See Annex A for a description of the non-standard device determination of cable type.

### 8.2.12 RESET- (Hardware reset)

This signal, referred to as hardware reset, shall be used by the host to reset the device (see 11.1).

> NOTE – While a minimum slew rate is not specified, some hosts assert RESET- with an extremely slow rise time when powering up. This may cause some devices to fail to recognize the assertion. Such hosts should negate then reassert RESET- once power has beenestablished.

### 8.2.13 CSEL (Cable select)

If CSEL is enabled in the device, the device is configured as either Device 0 or Device 1 depending upon the value of CSEL:

- If CSEL is negated, the device number is 0;
- If CSEL is asserted, the device number is 1.

The state of this signal may be sampled at any time by the device.

CSEL shall be grounded by the host.

#### 8.2.13.1 CSEL with 40-conductor cable

Special cabling may be used to selectively ground CSEL. CSEL of Device 0 is connected to the CSEL conductor in the cable, and is grounded, thus allowing the device to recognize itself as Device 0. CSEL of Device 1 is not connected to the CSEL conductor, thus the device recognizes itself as Device 1. If a single device is configured at the end of the cable using CSEL, a Device 1 only configuration results. See Figure 29 and Figure 30.

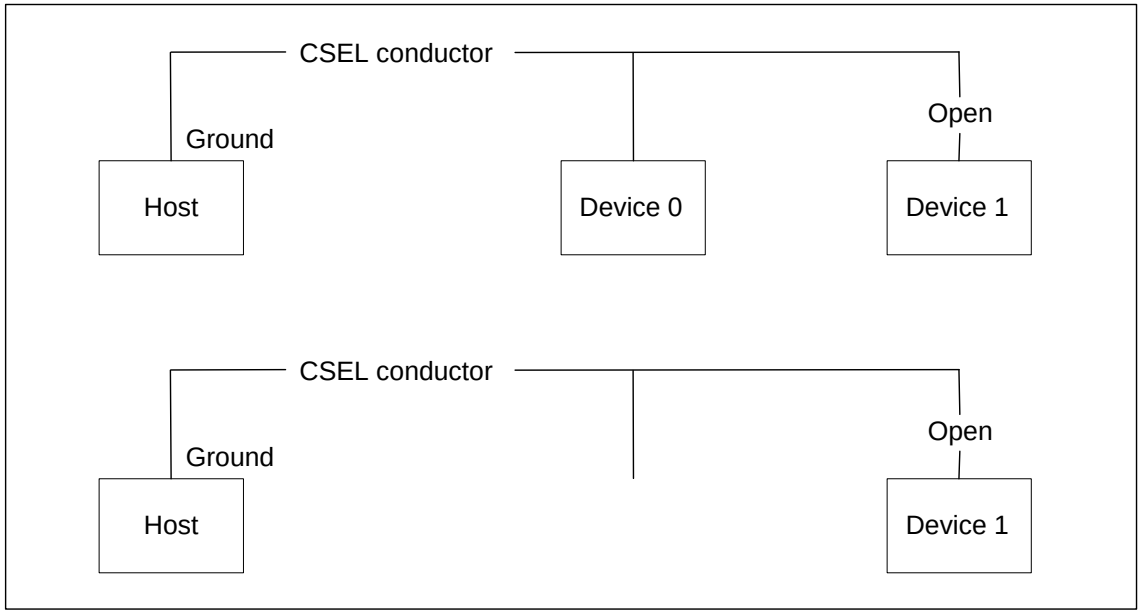



**Figure 29** – **Cable select example**

#### 8.2.13.2 CSEL with 80-conductor cable

For designated cable assemblies (including all 80-conductor cable assemblies): these assemblies are constructed so that CSEL is connected from the host connector to the connector at the opposite end of the cable from the host (see Figure 30). Therefore, Device 0 shall be at the opposite end of the cable from the host. Single device configurations with the device not at the end of the cable shall not be used with Ultra DMA modes.

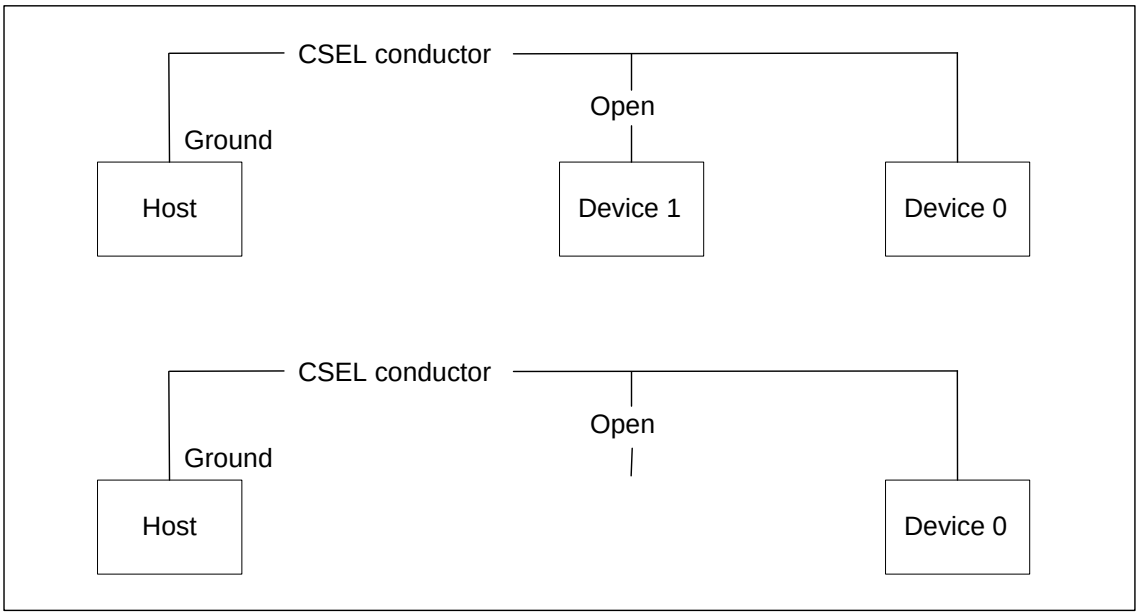



**Figure 30 – Alternate cable select example**

# 9 Parallel interface general operational requirements

## 9.1 Interrupts

INTRQ is used by the selected device to notify the host of an event. The device internal interrupt pending state is set when such an event occurs. If nIEN is cleared to zero, INTRQ is asserted (see 8.2.9).

The device shall enter the interrupt pending state when:

1) any command except a PIO data-in command reaches command completion successfully;
2) any command reaches command completion with error;
3) the device is ready to send a data block during a PIO data-in command;
4) the device is ready to accept a data block after the first data block during a PIO data-out command;
5) a device implementing the PACKET Command feature set is ready to receive the command packet and bits (6:5) in word 0 of the IDENTIFY PACKET DEVICE response have the value 01b;
6) a device implementing the PACKET Command feature set is ready to transfer a DRQ data block during a PIO transfer;
7) a device implementing the Overlap feature set performs a bus release if the bus release interrupt is enabled;
8) a device implementing the Overlap feature set has performed a bus release and is now ready to continue the command execution;
9) a device implementing the Overlap feature set is ready to transfer data after a SERVICE command if the Service interrupt is enabled;
10) Device 0 completes an EXECUTE DEVICE DIAGNOSTIC command. Device 1 shall not enter the interrupt pending state when completing an EXECUTE DEVICE DIAGNOSTIC command.

The device shall not exit the interrupt pending state as a result of the host changing the state of the DEV bit.

The device shall exit the interrupt pending state when:

1) the device is selected, BSY is cleared to zero, and the Status register is read;
2) the device is selected, both BSY and DRQ are cleared to zero, and the Command register is written;
3) the RESET- signal is asserted;
4) the SRST bit is set to one.

## 9.2 Multiword DMA

Multiword DMA is a mandatory data transfer protocol used with the READ DMA, READ DMA EXT, WRITE DMA, WRITE DMA EXT, READ DMA QUEUED, READ DMA QUEUED EXT, WRITE DMA QUEUED, WRITE DMA QUEUED EXT, and PACKET commands. When a Multiword DMA transfer is enabled as indicated by IDENTIFY DEVICE or IDENTIFY PACKET DEVICE (see clause 6) data, this data transfer protocol shall be used for the data transfers associated with these commands. DMA transfer modes may be changed using the SET FEATURES 03h subcommand (see clause 6). Signal timing for this protocol is described in 12.2.3.

The DMARQ and DMACK- signals are used to signify when a Multiword DMA transfer is to be executed. The DMARQ and DMACK- signals are also used to control the data flow of a Multiword DMA data transfer.

When a device is ready to transfer data associated with a Multiword DMA transfer, the device shall assert DMARQ. The host shall then respond by negating CS0- and CS1-, asserting DMACK-, and begin the data transfer by asserting, then negating, DIOW- or DIOR- for each word transferred. CS0- and CS1- shall remain negated as long as DMACK- is asserted. The host shall not assert DMACK- until DMARQ has been asserted by the device. The host shall initiate DMA read or write cycles only when both DMARQ and DMACK- are asserted. Having asserted DMARQ and DMACK-, these signals shall remain asserted until at least one word of data has been transferred.

The device may pause the transfer for flow control purposes by negating DMARQ. The host shall negate DMACK- in response to the negation of DMARQ. The device may then reassert DMARQ to continue the data transfer when the device is ready to transfer more data and DMACK- has been negated by the host.

The host may pause the transfer for flow control purposes by either pausing the assertion of DIOW- or DIOR- pulses or by negating DMACK-. The device may leave DMARQ asserted if DMACK- is negated. The host may then reassert DMACK- when DMARQ is asserted and begin asserting DIOW- or DIOR- pulses to continue the data transfer.

When the Multiword DMA data transfer is complete, the device shall negate DMARQ and the host shall negate DMACK- in response.

DMARQ shall be driven from the first assertion at the beginning of a DMA transfer until the negation after the last word is transferred. This signal shall be released at all other times.

If the device detects an error before data transfer for the command is complete, the device may complete the data transfer or may terminate the data transfer before completion and shall report the error in either case.

> NOTE – If a data transfer is terminated before completion, the assertion of INTRQ should be passed through to the host software driver regardless of whether all data requested by the command has been transferred.

## 9.3 Ultra DMA feature set

### 9.3.1 Overview

Ultra DMA is an optional data transfer protocol used with the READ DMA, READ DMA EXT, WRITE DMA, WRITE DMA EXT, READ DMA QUEUED, READ DMA QUEUED EXT, WRITE DMA QUEUED, WRITE DMA QUEUED EXT, and PACKET commands. When this protocol is enabled, the Ultra DMA protocol shall be used instead of the Multiword DMA protocol when these commands are issued by the host. This protocol applies to the Ultra DMA data burst only. When this protocol is used there are no changes to other elements of the ATA protocol (e.g., Command Block Register access).

Several signal lines are redefined to provide different functions during an Ultra DMA burst. These lines assume these definitions when:

1) an Ultra DMA mode is selected, and
2) a host issues a READ DMA, WRITE DMA, READ DMA QUEUED, WRITE DMA QUEUED, or a PACKET command requiring data transfer, and
3) the host asserts DMACK-.

These signal lines revert back to the definitions used for non-Ultra DMA transfers upon the negation of DMACK- by the host at the termination of an Ultra DMA burst.

With the Ultra DMA protocol, the STROBE signal that latches data from DD(15:0) is generated by the same agent (either host or device) that drives the data onto the bus. Ownership of DD(15:0) and this data strobe signal are given either to the device during an Ultra DMA data-in burst or to the host for an Ultra DMA data-out burst.

During an Ultra DMA burst a sender shall always drive data onto the bus, and, after a sufficient time to allow for propagation delay, cable settling, and setup time, the sender shall generate a STROBE edge to latch the data. Both edges of STROBE are used for data transfers so that the frequency of STROBE is limited to the same frequency as the data.

Words in the IDENTIFY DEVICE data indicate support of the Ultra DMA feature and the Ultra DMA modes the device is capable of supporting. The Set transfer mode subcommand in the SET FEATURES command shall be used by a host to select the Ultra DMA mode at which the system operates. The Ultra DMA mode selected by a host shall be less than or equal to the fastest mode of which the device is capable. Only one Ultra DMA mode shall be selected at any given time. All timing requirements for a selected Ultra DMA mode shall be satisfied. Devices supporting any Ultra DMA mode shall also support all slower Ultra DMA modes.

An Ultra DMA capable device shall retain the previously selected Ultra DMA mode after executing a software reset sequence or the sequence caused by receipt of a DEVICE RESET command if a SET FEATURES disable reverting to defaults command has been issued. The device may revert to a Multiword DMA mode if a SET FEATURES enable reverting to default has been issued. An Ultra DMA capable device shall clear any previously selected Ultra DMA mode and revert to the default non-Ultra DMA modes after executing a power-on or hardware reset.

Both the host and device perform a CRC function during an Ultra DMA burst. At the end of an Ultra DMA burst the host sends its CRC data to the device. The device compares its CRC data to the data sent from the host. If the two values do not match, the device reports an error in the error register. If an error occurs during one or more Ultra DMA bursts for any one command, the device shall report the first error that occurred. If the device detects that a CRC error has occurred before data transfer for the command is complete, the device may complete the transfer and report the error or abort the command and report the error.

NOTE – If a data transfer is terminated before completion, the assertion of INTRQ should be passed through to the host software driver regardless of whether all data requested by the command has been transferred.

### 9.3.2 Phases of operation

An Ultra DMA data transfer is accomplished through a series of Ultra DMA data-in or data-out bursts. Each Ultra DMA burst has three mandatory phases of operation: the initiation phase, the data transfer phase, and the Ultra DMA burst termination phase. In addition, an Ultra DMA burst may be paused during the data transfer phase (see 11.12 and 11.13 for the detailed protocol descriptions for each of these phases, 12.2.4 defines the specific timing requirements). In the following rules DMARDY- is used in cases that could apply to either DDMARDY- or HDMARDY-, and STROBE is used in cases that could apply to either DSTROBE or HSTROBE. The following are general Ultra DMA rules.

1) An Ultra DMA burst is defined as the period from an assertion of DMACK- by the host to the subsequent negation of DMACK-.

2) When operating in Ultra DMA modes 2, 1, or 0 a recipient shall be prepared to receive up to two data words whenever an Ultra DMA burst is paused. When operating in Ultra DMA modes 6, 5, 4, or 3 a recipient shall be prepared to receive up to three data words whenever an Ultra DMA burst is paused.

#### 9.3.2.1 Ultra DMA burst initiation phase rules

1) An Ultra DMA burst initiation phase begins with the assertion of DMARQ by a device and ends when the sender generates a STROBE edge to transfer the first data word.
2) An Ultra DMA burst shall always be requested by a device asserting DMARQ.
3) When ready to initiate the requested Ultra DMA burst, the host shall respond by asserting DMACK-.
4) A host shall never assert DMACK- without first detecting that DMARQ is asserted.
5) For Ultra DMA data-in bursts: a device may begin driving DD(15:0) after detecting that DMACK- is asserted, STOP negated, and HDMARDY- is asserted.
6) After asserting DMARQ or asserting DDMARDY- for an Ultra DMA data-out burst, a device shall not negate either signal until the first STROBE edge is generated.
7) After negating STOP or asserting HDMARDY- for an Ultra DMA data-in burst, a host shall not change the state of either signal until the first STROBE edge is generated.

#### 9.3.2.2 Data transfer phase rules

1) The data transfer phase is in effect from after Ultra DMA burst initiation until Ultra DMA burst termination.
2) A recipient pauses an Ultra DMA burst by negating DMARDY- and resumes an Ultra DMA burst by reasserting DMARDY-.
3) A sender pauses an Ultra DMA burst by not generating STROBE edges and resumes by generating STROBE edges.
4) A recipient shall not signal a termination request immediately when the sender stops generating STROBE edges. In the absence of a termination from the sender the recipient shall always negate DMARDY- and wait the required period before signaling a termination request.
5) A sender may generate STROBE edges at greater than the minimum period specified by the enabled Ultra DMA mode. The sender shall not generate STROBE edges at less than the minimum period specified by the enabled Ultra DMA mode. A recipient shall be able to receive data at the minimum period specified by the enabled Ultra DMA mode.

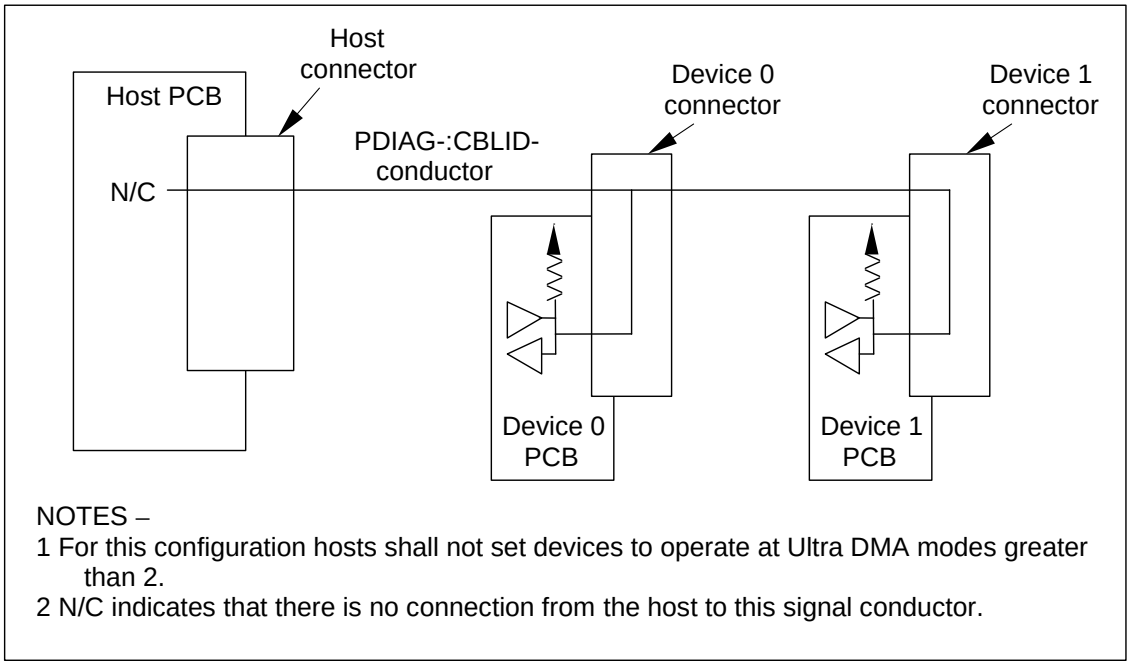
#### 9.3.2.3 Ultra DMA burst termination phase rules

1) Either a sender or a recipient may terminate an Ultra DMA burst.
2) Ultra DMA burst termination is not the same as command completion. If an Ultra DMA burst termination occurs before command completion, the command shall be completed by initiation of a new Ultra DMA burst at some later time or aborted by the host issuing a hardware or software reset or DEVICE RESET command if implemented by the device.
3) An Ultra DMA burst shall be paused before a recipient requests a termination.
4) A host requests a termination by asserting STOP. A device acknowledges a termination request by negating DMARQ.
5) A device requests a termination by negating DMARQ. A host acknowledges a termination request by asserting STOP.
6) Once a sender requests a termination, the sender shall not change the state of STROBE until the recipient acknowledges the request. Then, if STROBE is not in the asserted state, the sender shall return STROBE to the asserted state. No data shall be transferred on this transition of STROBE.
7) A sender shall return STROBE to the asserted state whenever the sender detects a termination request from the recipient. No data shall be transferred nor CRC calculated on this edge of DSTROBE.
8) Once a recipient requests a termination, the responder shall not change DMARDY from the negated state for the remainder of an Ultra DMA burst.
9) A recipient shall ignore a STROBE edge when DMARQ is negated or STOP is asserted.

## 9.4 Host determination of cable type by detecting CBLID-

In a system using a cable, hosts shall determine that an 80-conductor cable is installed in a system before operating with transfer modes faster than Ultra DMA mode 2. Hosts shall detect that CBLID- is connected to ground to determine the cable type. See 8.2.11.

For detecting that CBLID- is connected to ground, the host shall test to see if CBLID- is below $V_{IL}$ or above $V_{IH}$. If the signal is below $V_{IL}$, then an 80-conductor cable assembly is installed in the system because this signal is grounded in the 80-conductor cable assembly's host connector. If the signal is above $V_{IH}$, then a 40-conductor cable assembly is installed because this signal is connected to the device(s) and is pulled up through a 10 kΩ resistor at each device.




NOTES –
1 For this configuration hosts shall not set devices to operate at Ultra DMA modes greater than 2.
2 N/C indicates that there is no connection from the host to this signal conductor.

**Figure 31 - Example configuration of a system with a 40-conductor cable**

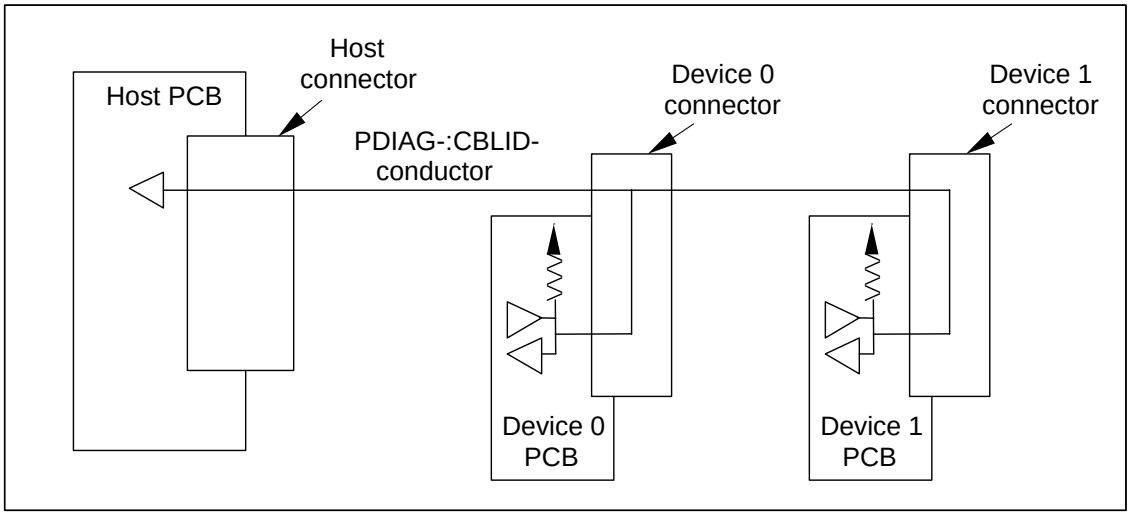



**Figure 32 - Example configuration of a system where the host detects a 40-conductor cable**

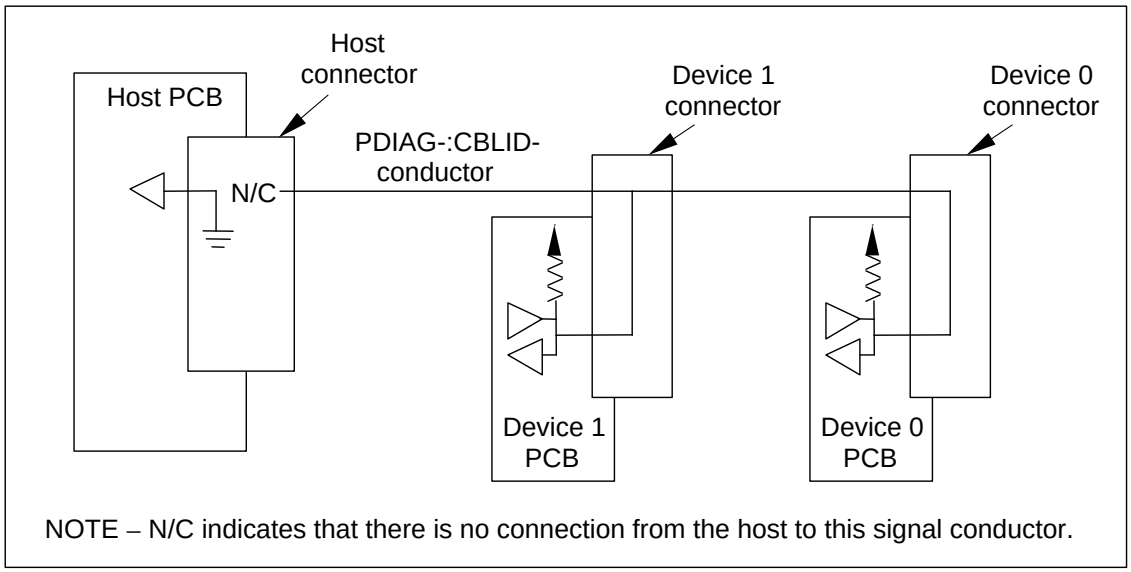



**Figure 33 - Example configuration of a system where the host detects an 80-conductor cable**

**Table 38 - Host detection of CBLID-**

| Cable assembly type | Device 1 releases PDIAG- | Electrical state of CBLID- at host | Host-determined cable type | Determination correct? |
|---|---|---|---|---|
| 40-conductor | Yes | 1 | 40-conductor | Yes |
| 80-conductor | Yes | 0 | 80-conductor | Yes |
| 40-conductor | No | 0 | 80-conductor | No (see note) |
| 80-conductor | No | 0 | 80-conductor | Yes |
| NOTE – Ultra DMA mode 3, 4, 5, or 6 may be set incorrectly resulting in ICRC errors. | | | | |

## 10 Parallel interface register addressing

Registers are defined depending on whether reading or writing the register and whether the PACKET command feature set is implemented (see Volume 1).

**Table 39 – I/O registers**

| **Registers used by devices not implementing the PACKET command feature set** | | **Registers used by devices implementing the PACKET command feature set** | |
|---|---|---|---|
| **Command Block registers** | | **Command Block registers** | |
| **When read** | **When written** | **When read** | **When written** |
| Data | Data | Data | Data |
| Error | Features | Error | Features |
| Sector Count | Sector Count | Interrupt reason | |
| LBA Low | LBA Low | | |
| LBA Mid | LBA Mid | Byte Count Low | Byte Count Low |
| LBA High | LBA High | Byte Count High | Byte Count High |
| Device | Device | Device select | Device select |
| Status | Command | Status | Command |
| **Control Block registers** | | **Control Block registers** | |
| Alternate Status | Device Control | Alternate Status | Device Control |

For transport protocols and timing see clauses 11 and 12.

When the host initiates a register or Data port read or write cycle by asserting then negating either DIOW- or DIOR-, the device(s) shall determine how to respond and what action(s), if any, are to be taken. The following text and tables describe this decision process.

The device response begins with these steps:

1) For a device that is not in Sleep mode, see Table 40.
2) If DMACK- is asserted, a device in Sleep mode shall ignore all DIOW-/DIOR- activity. If DMACK- is not asserted, a device in Sleep mode shall respond as described in Table 45 if the device does not implement the PACKET Command feature set or Table 46 if the device does implement the PACKET Command feature set.

**Table 40 – Device response to DIOW-/DIOR-**

| **Is the device selected?** (see note 1) | **Is DMACK- asserted?** | **Action/Response** |
|---|---|---|
| No | No | See Table 41 |
| No | Yes | DIOW-/DIOR- cycle is ignored (possible DMA transfer with the other device) |
| Yes | No | See Table 42 |
| Yes | Yes | See Table 43 (see note 2) |
| Device 1 is selected but there is no Device 1 and Device 0 responds for Device 1. | No | See Table 44 and clause 5. |
| Device 1 is selected but there is no Device 1 and Device 0 responds for Device 1. | Yes | DIOW-/DIOR- cycle is ignored (possible malfunction of the host) |
| NOTES –<br>1 Device selected means that the DEV bit in the Device register matches the logical device number of the device.<br>2 Applicable only to Multiword DMA, not applicable to Ultra DMA. | | |

**Table 41 – Device is not selected, DMACK- is not asserted**

| CS0- | CS1- | DA2 | DA1 | DA0 | DIOx- | DMARQ | BSY | DRQ | Device Response |
|---|---|---|---|---|---|---|---|---|---|
| N | N | X | X | X | X | Z | X | X | DIOW-/DIOR- cycle is ignored. |
| N | A | N | X | X | X | Z | X | X | |
| N | A | A | N | X | X | Z | X | X | |
| N | A | A | A | N | W | Z | X | X | Place new data into the Device Control register and respond to the new values of the nIEN and SRST bits. |
| N | A | A | A | N | R | Z | X | X | DIOW-/DIOR- cycle is ignored. |
| N | A | A | A | A | X | Z | X | X | |
| A | N | N | N | N | X | Z | X | X | |
| A | N | N | N | A | W | Z | 0 | X | Place new data into the Feature register. |
| A | N | N | N | A | W | Z | 1 | X | DIOW-/DIOR- cycle is ignored. |
| A | N | N | N | A | R | Z | X | X | |
| A | N | N | A | N | W | Z | 0 | X | Place new data into the Sector Count register. |
| A | N | N | A | N | W | Z | 1 | X | DIOW-/DIOR- cycle is ignored. |
| A | N | N | A | N | R | Z | X | X | |
| A | N | N | A | A | W | Z | 0 | X | Place new data into the LBA Low register. |
| A | N | N | A | A | W | Z | 1 | X | DIOW-/DIOR- cycle is ignored. |
| A | N | N | A | A | R | Z | X | X | |
| A | N | A | N | N | W | Z | 0 | X | Place new data into the LBA Mid register. |
| A | N | A | N | N | W | Z | 1 | X | DIOW-/DIOR- cycle is ignored. |
| A | N | A | N | N | R | Z | X | X | |
| A | N | A | N | A | W | Z | 0 | X | Place new data into the LBA High register. |
| A | N | A | N | A | W | Z | 1 | X | DIOW-/DIOR- cycle is ignored. |
| A | N | A | N | A | R | Z | X | X | |
| A | N | A | A | N | W | Z | 0 | X | Place new data into the Device register. Respond to the new value of the DEV bit. |
| A | N | A | A | N | W | Z | 1 | X | DIOW-/DIOR- cycle is ignored. |
| A | N | A | A | N | R | Z | X | X | |
| A | N | A | A | A | W | Z | 0 | X | Place new data into the Command register. Do not respond unless the command is EXECUTE DEVICE DIAGNOSTICS. |
| A | N | A | A | A | W | Z | 1 | X | DIOW-/DIOR- cycle is ignored. |
| A | N | A | A | A | R | Z | X | X | |
| A | A | X | X | X | X | Z | X | X | DIOW-/DIOR- cycle is ignored. |

NOTE –
1. Except in the DIOx- column, A = asserted, N = negated, Z = released, X = don't care.
2. In the DIOx- column, R = DIOR- asserted, W = DIOW- asserted, X = either DIOR- or DIOW- is asserted.
3. Device is selected if the DEV bit in the Device register is the logical device number of the device.

**Table 42 – Device is selected, DMACK- is not asserted**

| CS0- | CS1- | DA2 | DA1 | DA0 | DIOx- | DMARQ | BSY | DRQ | Device Response |
|---|---|---|---|---|---|---|---|---|---|
| N<br>N<br>N | N<br>A<br>A | X<br>N<br>A | X<br>X<br>N | X<br>X<br>X | X<br>X<br>X | X<br>X<br>X | X<br>X<br>X | X<br>X<br>X | DIOW-/DIOR- cycle is ignored. |
| N | A | A | A | N | W | X | X | X | Place new data into the Device Control register and respond to the new values of the nIEN and SRST bits. |
| N | A | A | A | N | R | X | X | X | Place Status register contents on the data bus (do not change the Interrupt Pending state). |
| N<br>A | A<br>N | A<br>N | A<br>N | A<br>N | X<br>X | X<br>X | X<br>0 | X<br>0 | DIOW-/DIOR- cycle is ignored. |
| A | N | N | N | N | X | X | 0 | 1 | PIO data transfer for this device, a 16-bit data word is transferred via the Data register. |
| A | N | N | N | N | X | X | 1 | X | Result of DIOW-/DIOR- cycle is indeterminate. |
| A | N | N | N | A | W | X | 0 | 0 | Place new data into the Features register. |
| A | N | N | N | A | W | X | 0 | 1 | DIOW- is ignored, this is a malfunction of the host. |
| A | N | N | N | A | W | X | 1 | X | Result of DIOW-/DIOR- cycle is indeterminate. |
| A | N | N | N | A | R | X | 0 | X | Place the contents of the Error register on the data bus. |
| A | N | N | N | A | R | X | 1 | X | Place the contents of the Status register on the data bus. |
| A | N | N | A | N | W | X | 0 | 0 | Place new data into the Sector Count register. |
| A | N | N | A | N | W | X | 0 | 1 | DIOW- is ignored, this is a malfunction of the host. |
| A | N | N | A | N | W | X | 1 | X | Result of DIOW-/DIOR- cycle is indeterminate. |
| A | N | N | A | N | R | X | 0 | X | Place the contents of the Sector Count register on the data bus. |
| A | N | N | A | N | R | X | 1 | X | Place the contents of the Status register on the data bus. |
| A | N | N | A | A | W | X | 0 | 0 | Place new data into the LBA Low register. |
| A | N | N | A | A | W | X | 0 | 1 | DIOW- is ignored, this is a malfunction of the host. |
| A | N | N | A | A | W | X | 1 | X | Result of DIOW-/DIOR- cycle is indeterminate. |
| A | N | N | A | A | R | X | 0 | X | Place the contents of the LBA Low register on the data bus. |
| A | N | N | A | A | R | X | 1 | X | Place the contents of the Status register on the data bus. |
| A | N | A | N | N | W | X | 0 | 0 | Place new data into the LBA Mid register. |
| A | N | A | N | N | W | X | 0 | 1 | DIOW- is ignored, this is a malfunction of the host. |
| A | N | A | N | N | W | X | 1 | X | Result of DIOW-/DIOR- cycle is indeterminate. |
| A | N | A | N | N | R | X | 0 | X | Place the contents of the LBA Mid register on the data bus. |
| A | N | A | N | N | R | X | 1 | X | Place the contents of the Status register on the data bus. |
| A | N | A | N | A | W | X | 0 | 0 | Place new data into the LBA High register. |

*(continued)*

**Table 42 – Device is selected, DMACK- is not asserted** *(continued)*

| CS0- | CS1- | DA2 | DA1 | DA0 | DIOx- | DMARQ | BSY | DRQ | Device Response |
|---|---|---|---|---|---|---|---|---|---|
| A | N | A | N | A | W | X | 0 | 1 | DIOW- is ignored, this is a malfunction of the host. |
| A | N | A | N | A | W | X | 1 | X | Result of DIOW-/DIOR- cycle is indeterminate. |
| A | N | A | N | A | R | X | 0 | X | Place the contents of the LBA High register on the data bus. |
| A | N | A | N | A | R | X | 1 | X | Place the contents of the Status register on the data bus. |
| A | N | A | A | N | W | X | 0 | 0 | Place new data into the Device register. Respond to the new value of the DEV bit. |
| A | N | A | A | N | W | X | 0 | 1 | DIOW- is ignored, this is a malfunction of the host. |
| A | N | A | A | N | W | X | 1 | X | Result of DIOW-/DIOR- cycle is indeterminate. |
| A | N | A | A | N | R | X | 0 | X | Place the contents of the Device register on the data bus. |
| A | N | A | A | N | R | X | 1 | X | Place the contents of the Status register on the data bus. |
| A | N | A | A | A | W | X | 0 | 0 | Place new data into the Command register and respond to the new command (exit the interrupt pending State). |
| A<br>A | N<br>N | A<br>A | A<br>A | A<br>A | W<br>W | X<br>X | 0<br>1 | 1<br>X | Result of DIOW-/DIOR- cycle is indeterminate, unless the device supports DEVICE RESET. If the device supports the DEVICE RESET command, exit the interrupt pending state. |
| A | N | A | A | A | R | X | X | X | Place contents of Status register on the data bus and exit the interrupt pending state. |
| A | A | X | X | X | X | X | X | X | DIOW-/DIOR- cycle is ignored. |

NOTE –
1. Except in the DIOx- column, A = asserted, N = negated, X = don't care.
2. In the DIOx- column, R = DIOR- asserted, W = DIOW- asserted, X = either DIOR- or DIOW- is asserted.
3. Device is selected if the DEV bit in the Device register is the logical device number of the device.
4. For devices implementing the 48-bit Address feature set, the HOB bit in the Device Control register defines whether the current or previous content of the registers is placed on DD(7:0).

*(concluded)*

**Table 43 – Device is selected, DMACK- is asserted (for Multiword DMA only)**

| CS0- | CS1- | DA2 | DA1 | DA0 | DIOx- | DMARQ | BSY | DRQ | Device Response |
|---|---|---|---|---|---|---|---|---|---|
| N<br>N | N<br>N | X<br>X | X<br>X | X<br>X | X<br>X | N<br>N | 1<br>0 | X<br>X | This could be the final DIOW-/DIOR- of a Multiword DMA transfer burst, or a possible malfunction of the host that is ignored. |
| N<br>N | N<br>N | X<br>X | X<br>X | X<br>X | X<br>X | A<br>A | 1<br>0 | X<br>1 | DMA transfer for this device, a 16-bit word of data is transferred via the Data Port. |
| X<br>A | A<br>X | X<br>X | X<br>X | X<br>X | X<br>X | X<br>X | X<br>X | X<br>X | DIOW-/DIOR- cycle is ignored (possible malfunction of the host). |

NOTE –
1. Except in the DIOx- column, A = asserted, N = negated, Z = released, X = don't care.
2. In the DIOx- column, R = DIOR- asserted, W = DIOW- asserted, X = either DIOR- or DIOW- is asserted.
3. Device is selected if the DEV bit in the Device register is the logical device number of the device.

**Table 44 –Device 1 is selected and Device 0 is responding for Device 1**

| CS0- | CS1- | DA2 | DA1 | DA0 | DIOx- | DMARQ | BSY | DRQ | Device Response |
|---|---|---|---|---|---|---|---|---|---|
| N<br>N<br>N | N<br>A<br>A | X<br>N<br>A | X<br>X<br>N | X<br>X<br>X | X<br>X<br>X | Z<br>Z<br>X | 0<br>0<br>0 | 0<br>0<br>0 | DIOW-/DIOR- cycle is ignored. |
| N | A | A | A | N | W | X | 0 | 0 | Place new data into the Device 0 Device Control register and respond to the new values of the nIEN and SRST bits. |
| N | A | A | A | N | R | X | 0 | 0 | Place 00H on the data bus. |
| N<br>A | A<br>N | A<br>N | A<br>N | A<br>N | X<br>X | X<br>X | 0<br>0 | 0<br>0 | DIOW-/DIOR- cycle is ignored. |
| A | N | N | N | A | W | X | 0 | 0 | Place new data into the Device 0 Feature register. |
| A | N | N | N | A | R | X | 0 | 0 | Place the contents of the Device 0 Error register on the data bus. |
| A | N | N | A | N | W | X | 0 | 0 | Place new data into Device 0 Sector Count register. |
| A | N | N | A | N | R | X | 0 | 0 | If the device does not implement the PACKET Command feature set, the device shall place the contents of the Device 0 Sector Count register on the data bus. If the device implements the PACKET Command feature set, the device shall place 00h on the data bus. |
| A | N | N | A | A | W | X | 0 | 0 | Place new data into Device 0 LBA Low register. |
| A | N | N | A | A | R | X | 0 | 0 | If the device does not implement the PACKET Command feature set, the device shall place the contents of the Device 0 LBA Low register on the data bus. If the device implements the PACKET Command feature set, the device shall place 00h on the data bus. |
| A | N | A | N | N | W | X | 0 | 0 | Place new data into Device 0 LBA Mid register. |
| A | N | A | N | N | R | X | 0 | 0 | If the device does not implement the PACKET Command feature set, the device shall place the contents of the Device 0 LBA Mid register on the data bus. If the device implements the PACKET Command feature set, the device shall place 00h on the data bus. |
| A | N | A | N | A | W | X | 0 | 0 | Place new data into Device 0 LBA High register. |
| A | N | A | N | A | R | X | 0 | 0 | If the device does not implement the PACKET Command feature set, the device shall place the contents of the Device 0 LBA High register on the data bus. If the device implements the PACKET Command feature set, the device shall place 00h on the data bus. |
| A | N | A | A | N | W | X | 0 | 0 | Place new data into the Device 0 Device register. Respond to the new value of the DEV bit. |

*(continued)*

**Table 44 –Device 1 is selected and Device 0 is responding for Device 1** *(continued)*

| CS0- | CS1- | DA2 | DA1 | DA0 | DIOx- | DMARQ | BSY | DRQ | Device Response |
|---|---|---|---|---|---|---|---|---|---|
| A | N | A | A | N | R | X | 0 | 0 | If the device does not implement the PACKET Command feature set, the device shall place the contents of the Device 0 Device register, with the DEV bit set to one, on the data bus. If the device implements the PACKET Command feature set, the device shall place 00h on the data bus. |
| A | N | A | A | A | W | X | 0 | 0 | Place new data into the Command register of Device 0. Do not respond unless the command is EXECUTE DEVICE DIAGNOSTICS. |
| A | N | A | A | A | R | X | 0 | 0 | Place 00H on the data bus. |
| A | A | X | X | X | X | X | 0 | 0 | DIOW-/DIOR- cycle is ignored. |

NOTE –
1. Except in the DIOx- column, A = asserted, N = negated, Z = released, X = don't care.
2. In the DIOx- column, R = DIOR- asserted, W = DIOW- asserted, X = either DIOR- or DIOW- is asserted.
3. Device is selected if the DEV bit in the Device register is the logical device number of the device.

*(concluded)*

**Table 45 – Device is in Sleep mode, DEVICE RESET is not implemented, DMACK- is not asserted**

| CS0- | CS1- | DA2 | DA1 | DA0 | DIOx- | DMARQ | BSY | DRQ | Device Response |
|---|---|---|---|---|---|---|---|---|---|
| N | N | X | X | X | X | Z | X | X | DIOW-/DIOR- cycle is ignored. |
| N | A | N | X | X | X | Z | X | X | |
| N | A | A | N | X | X | Z | X | X | |
| N | A | A | A | N | W | Z | X | X | Place new data into the Device Control register SRST bit and respond only if SRST bit is 1. |
| N | A | A | A | N | R | Z | X | X | DIOW-/DIOR- cycle is ignored. |
| N | A | A | A | A | X | Z | X | X | |
| A | N | X | X | X | X | Z | X | X | |
| A | A | X | X | X | X | Z | X | X | |

NOTE –
1. Except in the DIOx- column, A = asserted, N = negated, Z = released, X = don't care.
2. In the DIOx- column, R = DIOR- asserted, W = DIOW- asserted, X = either DIOR- or DIOW- is asserted.
3. Device is selected if the DEV bit in the Device register is the logical device number of the device.

**Table 46 – Device is in Sleep mode, DEVICE RESET is implemented, DMACK- is not asserted**

| CS0- | CS1- | DA2 | DA1 | DA0 | DIOx- | DMARQ | BSY | DRQ | Device Response |
|---|---|---|---|---|---|---|---|---|---|
| N | N | X | X | X | X | Z | X | X | DIOW-/DIOR- cycle is ignored. |
| N | A | N | X | X | X | Z | X | X | |
| N | A | A | N | X | X | Z | X | X | |
| N | A | A | A | N | W | Z | X | X | Place new data into the Device Control register SRST bit and respond only if SRST bit is 1. |
| N | A | A | A | N | R | Z | X | X | DIOW-/DIOR- cycle is ignored. |
| N | A | A | A | A | X | Z | X | X | |
| A | N | N | X | X | X | Z | X | X | |
| A | N | A | N | X | X | Z | X | X | |
| A | N | A | A | N | W | Z | X | X | Place new data into the Device register DEV bit. |
| A | N | A | A | N | R | Z | X | X | DIOR- cycle is ignored. |
| A | N | A | A | A | W | Z | X | X | DIOW- cycle is ignored unless the device is selected and the command is DEVICE RESET. |
| A | N | A | A | A | R | Z | X | X | DIOR- cycle is ignored. |
| A | A | X | X | X | X | Z | X | X | DIOW-/DIOR- cycle is ignored |

NOTE –
1. Except in the DIOx- column, A = asserted, N = negated, Z = released, X = don't care.
2. In the DIOx- column, R = DIOR- asserted, W = DIOW- asserted, X = either DIOR- or DIOW- is asserted.
3. Device is selected if the DEV bit in the Device register is the logical device number of the device.

# 11 Parallel interface transport protocol

Commands are grouped into different classes according to the protocol followed for command execution. The command classes with their associated protocol are defined in state diagrams in this clause, one state diagram for host actions and a second state diagram for device actions. Figure 34 shows the overall relationship of the host protocol state diagrams. Figure 35 shows the overall relationship of the device protocol state diagrams. State diagrams defining these protocols are not normative descriptions of implementations, they are normative descriptions of externally apparent device or host behavior. Different implementations are allowed. See 3.2.7 for state diagram conventions.

A device shall not timeout any activity when waiting for a response from the host.

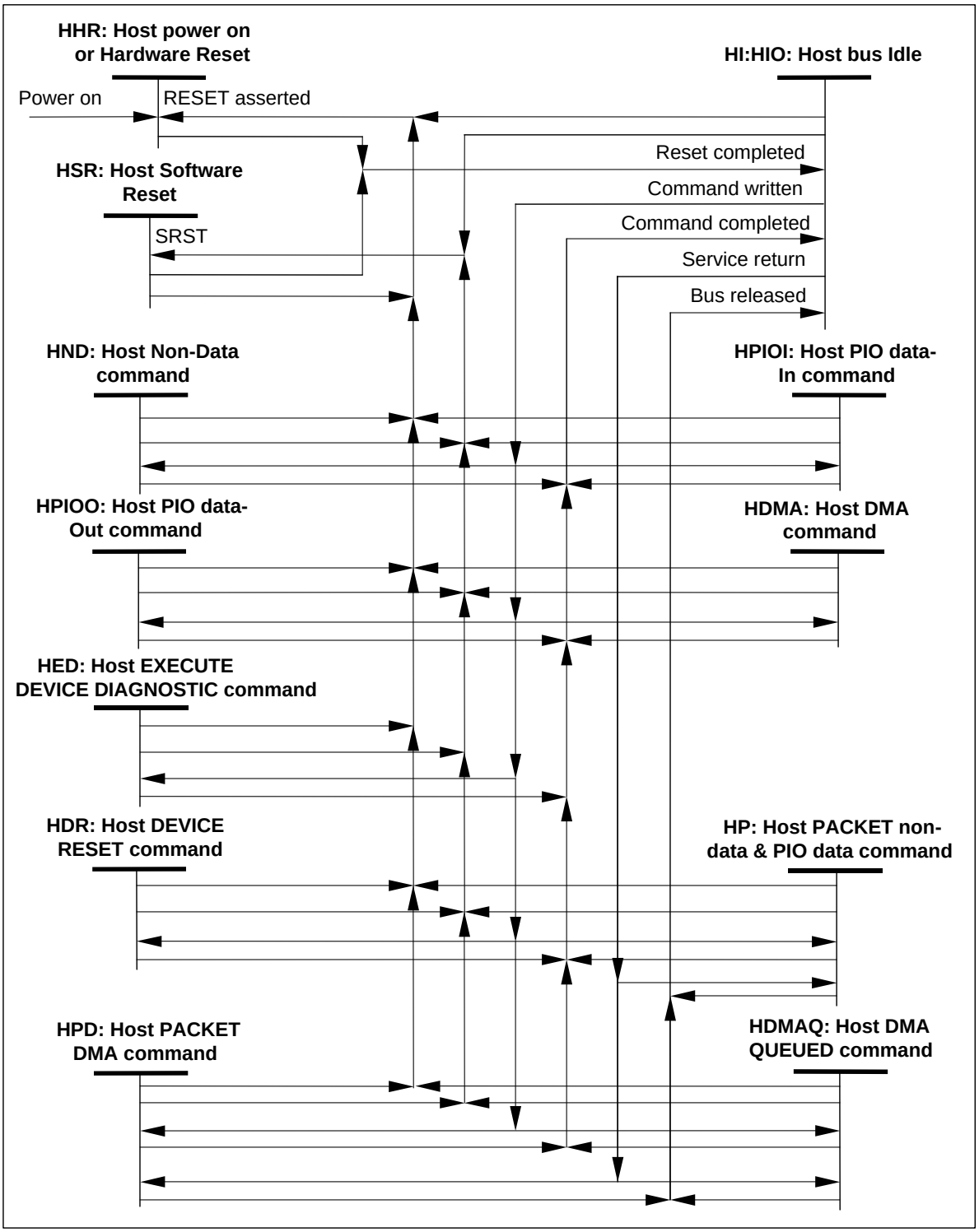



**Figure 34 – Overall host protocol state sequence**

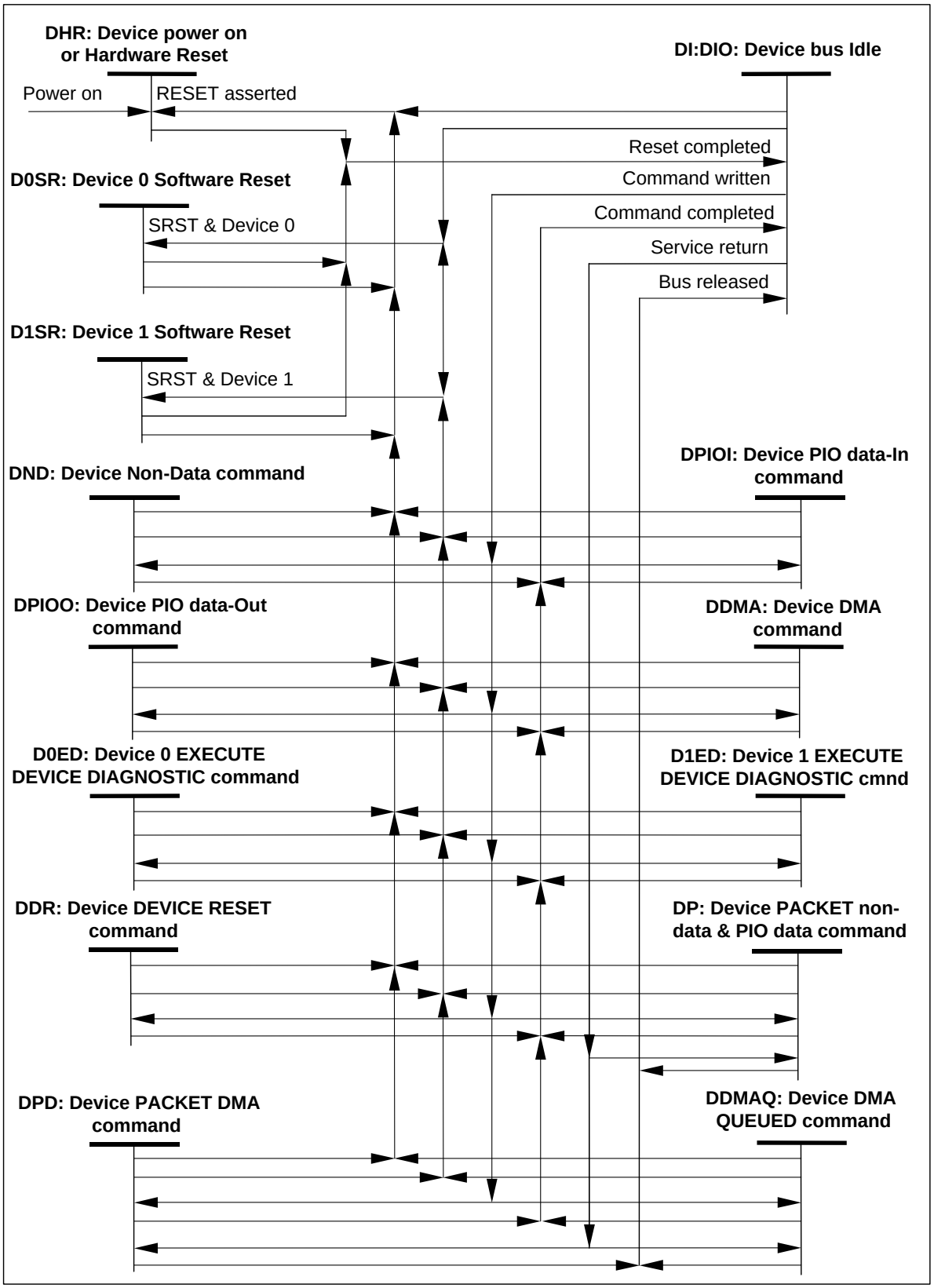



**Figure 35 – Overall device protocol state sequence**

## 11.1 Power-on and hardware reset protocol

This clause describes the protocol for processing of power-on and hardware resets.

If the host asserts RESET-, regardless of the power management mode, the device shall execute the hardware reset protocol. If the host reasserts RESET- before a device has completed the power-on or hardware reset protocol, then the device shall restart the protocol from the begining.

The host should not set the SRST bit to one in the Device Control register or issue a DEVICE RESET command while the BSY bit is set to one in either device Status register as a result of executing the power-on or hardware reset protocol. If the host sets the SRST bit in the Device Control register to one or issues a DEVICE RESET command before devices have completed execution of the power-on or hardware reset protocol, then the devices shall ignore the software reset or DEVICE RESET command.

A host should issue an IDENTIFY DEVICE and/or IDENTIFY PACKET DEVICE command after the power-on or hardware reset protocol has completed to determine the current status of features implemented by the device(s).

Note: Serial implementations of ATA have different hardware reset timeout requirements, see Volume 3.

Figure 36 and the text following the figure decribes the power-on or hardware reset protocol for the host. Figure 37 and the text following the figure decribes the power-on or hardware reset protocol for the devices.

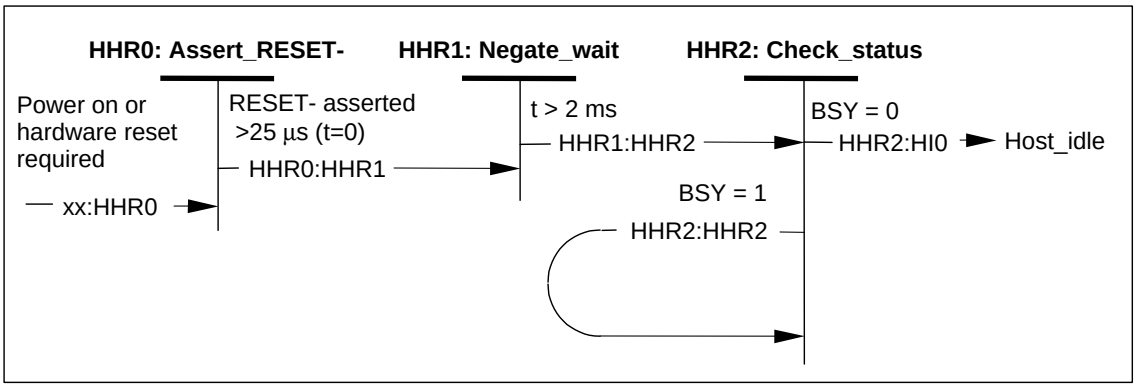



**Figure 36 – Host power-on or hardware reset state diagram**

**HHR0: Assert_RESET- State:** This state is entered at power-on or when the host recognizes that a hardware reset is required.

When in this state, the host asserts RESET-. The host shall remain in this state with RESET- asserted for at least 25 µs. The host shall negate CS(1:0), DA(2:0), DMACK-, DIOR-, and DIOW- and release DD(15:0).

**Transition HHR0:HHR1:** When the host has had RESET- asserted for at least 25 µs, the host shall make a transition to the HHR1: Negate_wait state.

**HHR1: Negate_wait State:** This state is entered when RESET- has been asserted for at least 25 µs.

When in this state, the host shall negate RESET-. The host shall remain in this state for at least 2 ms after negating RESET-. If the host tests CBLID- it shall do so at this time.

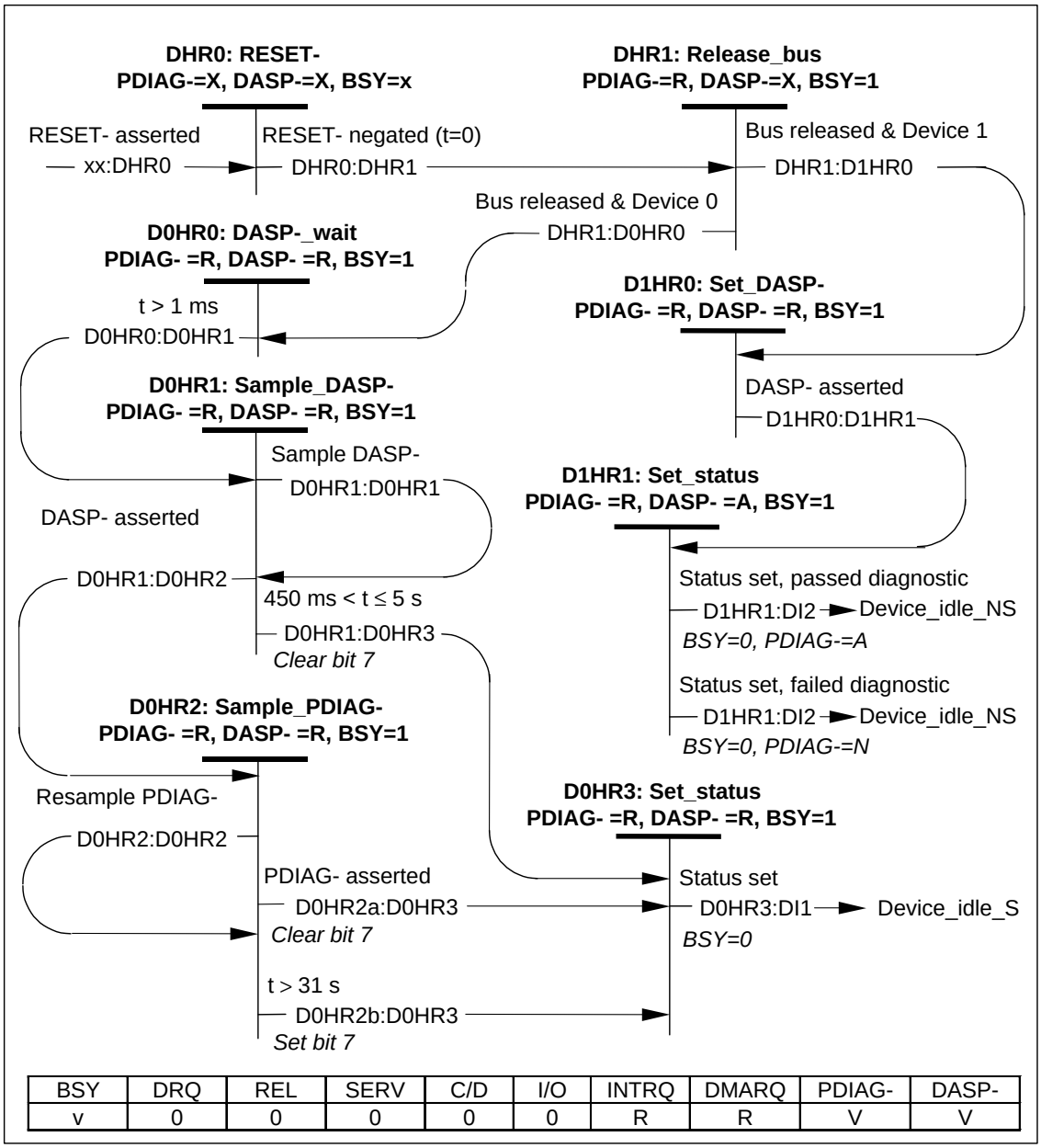
**Transition HHR1:HHR2:** When RESET- has been negated for at least 2 ms, the host shall make a transition to the HHR2: Check_status state.

**HHR2: Check_status State:** This state is entered when RESET- has been negated for at least 2 ms.

When in this state the host shall read the Status or Alternate Status register.

**Transition HHR2:HHR2:** When BSY is set to one, the host shall make a transition to the HHR2: Check_status state.

**Transition HHR2:HI0:** When BSY is cleared to zero, the host shall make a transition to the HI0: Host_idle state (see Figure 41). If status indicates that an error has occurred, the host shall take appropriate error recovery action.




| BSY | DRQ | REL | SERV | C/D | I/O | INTRQ | DMARQ | PDIAG- | DASP- |
|---|---|---|---|---|---|---|---|---|---|
| v | 0 | 0 | 0 | 0 | 0 | R | R | V | V |

**Figure 37 – Device power-on or hardware reset state diagram**

**DHR0: RESET State:** This state is entered when a valid assertion of the RESET- signal is recognized. The device shall not recognize a RESET- assertion shorter than 20 ns as valid. Devices may recognize a RESET- assertion greater that 20 ns as valid and shall recognize a RESET- assertion equal to or greater than 25 μs as valid.

**Transition DHR0:DHR1:** When a valid RESET- signal is negated, the device shall make a transition to the DHR1: Release_Bus state.

**DHR1: Release_bus State:** This state is entered when a valid RESET- signal is negated.

When in this state, the device shall release bus signals PDIAG-, INTRQ, IORDY, DMARQ, and DD(15:0) and shall set BSY to one within 400 ns after entering this state. The device shall determine if the device is Device 0 or Device 1 by checking the jumper, switch, or CSEL.

**Transition DHR1:D0HR0:** When the device has determined that the device is Device 0, has released the bus signals, and has set BSY to one, then the device shall make a transition to the D0HR0: DASP-_wait state.

**Transition DHR1:D1HR0:** When the device has determined that the device is Device 1, has released the bus signals, and has set BSY to one, then the device shall make a transition to the D1HR0: Set_DASP- state.

**D0HR0: DASP-_wait State:** This state is entered when the device has released the bus signals, set BSY to one, and determined that the device is Device 0.

When in this state, the device shall release DASP- and clear the DEV bit in the Device register to zero within 1 ms of the negation of RESET-.

**Transition D0HR0:D0HR1:** When at least 1 ms has elapsed since the negation of RESET-, the device shall make a transition to the D0HR1: Sample_DASP- state.

**D0HR1: Sample_DASP- State:** This state is entered when at least 1 ms has elapsed since the negation of RESET-.

When in this state, the device should begin performing the hardware initialization and self-diagnostic testing. This may revert the device to the default condition (the device's settings may now be different than they were before the host asserted RESET-). All Ultra DMA modes shall be disabled.

When in this state, the device shall sample the DASP- signal.

**Transition D0HR1:D0HR2:** When the sample indicates that DASP- is asserted, the device shall make a transition to the D0HR2: Sample_PDIAG- state.

**Transition D0HR1:D0HR1:** When the sample indicates that DASP- is negated and less than 450 ms have elapsed since the negation of RESET-, then the device shall make a transition to the D0HR1: Sample_DASP- state. When the sample indicates that DASP- is negated and greater than 450 ms but less than 5 s have elapsed since the negation of RESET-, then the device may make a transition to the D0HR1: Sample_DASP- state.

**Transition D0HR1:D0HR3:** When the sample indicates that DASP- is negated and 5 s have elapsed since the negation of RESET-, then the device shall clear bit 7 in the Error register and make a transition to the D0HR3: Set_status state. When the sample indicates that DASP- is negated and greater than 450 ms but less than 5 s have elapsed since the negation of RESET-, then the device may clear bit 7 in the Error register and make a transition to the D0HR3: Set_status state.

**D0HR2: Sample_PDIAG- State:** This state is entered when the device has recognized that DASP- is asserted.

When in this state, the device shall sample the PDIAG- signal.

**Transition D0HR2a:D0HR3:** When the sample indicates that PDIAG- is asserted, the device shall clear bit 7 in the Error register and make a transition to the D0HR3: Set_status state.

**Transition D0HR2b:D0HR3:** When the sample indicates that PDIAG- is not asserted and 31 s have elapsed since the negation of RESET-, then the device shall set bit 7 in the Error register and make a transition to the D0HR3: Set_status state.

**Transition D0HR2:D0HR2:** When the sample indicates that PDIAG- is not asserted and less than 31 s have elapsed since the negation of RESET-, then the device shall make a transition to the D0HR2: Sample_PDIAG- state.

**D0HR3: Set_status State:** This state is entered when Bit 7 in the Error register has been set or cleared.

When in this state the device shall complete the hardware initialization and self-diagnostic testing begun in the Sample DASP- state if not already completed.

The EXECUTE DEVICE DIAGNOSTICS diagnostic code shall be placed in bits (6:0) of the Error register (see clause 6). The device shall set the signature values (see clause 5). The device shall clear the SRST bit to zero in the Device Control register if set set to one. The content of the Features register is undefined. The device shall set word 93 in the IDENTIFY DEVICE or IDENTIFY PACKET DEVICE response (see clause 6).

If the device does not implement the PACKET command feature set, the device shall clear bits 3, 2, and 0 in the Status register to zero.

If the device implements the PACKET command feature set, the device shall clear bits 6, 5, 4, 3, 2, and 0 in the Status register to zero. The device shall return the operating modes to their specified initial conditions. MODE SELECT conditions shall be restored to their last saved values if saved values have been established. MODE SELECT conditions for which no values have been saved shall be returned to their default values.

**Transition D0HR3:DI1:** When hardware initialization and self-diagnostic testing is completed and the status has been set, the device shall clear BSY to zero and make a transition to the DI1: Device_idle_S state (see Figure 43).

**D1HR0: Set_DASP- State:** This state is entered when the device has released the bus, set BSY to one, and determined that the device is Device 1.

When in this state, the device shall clear the DEV bit in the Device register to zero within 1 ms and shall assert DASP- within 400 ms of the negation of RESET-.

When in this state, the device should begin execution of the hardware initialization and self-diagnostic testing. The device may revert to the default condition (the device's settings may now be in different conditions than they were before RESET- was asserted by the host). All Ultra DMA modes shall be disabled.

**Transition D1HR0:D1HR1:** When DASP- has been asserted, the device shall make a transition to the D1HR1: Set_status state.

**D1HR1: Set_status State:** This state is entered when the device has asserted DASP-.

When in this state the device shall complete any hardware initialization and self-diagnostic testing begun in the Set DASP- state if not already completed. The EXECUTE DEVICE DIAGNOSTIC diagnostic code shall be placed in the Error register (see clause 6). If the device passed self-diagnostics, the device shall assert PDIAG-. The device shall set word 93 in the IDENTIFY DEVICE or IDENTIFY PACKET DEVICE response (see clause 6).

All actions required in this state shall be completed in ≤ 30 s.

The device shall set the signature values (see clause 5). The content of the Features register is undefined. The device shall clear the SRST bit to zero in the Device Control register if set set to one.

If the device does not implement the PACKET command feature set, the device shall clear bits 3, 2, and 0 in the Status register to zero.

If the device implements the PACKET command feature set, the device shall clear bits 6, 5, 4, 3, 2, and 0 in the Status register to zero. The device shall return the operating modes to their specified initial conditions. MODE SELECT conditions shall be restored to their last saved values if saved values have been established. MODE SELECT conditions for which no values have been saved shall be returned to their default values.

**Transition D1HR1a:DI2:** When hardware initialization and self-diagnostic testing is completed, the device passed its diagnostics, and the status has been set, the device shall clear BSY to zero, assert PDIAG-, and make a transition to the DI2: Device_idle_NS state (see Figure 43).

**Transition D1HR1b:DI2:** When hardware initialization and self-diagnostic testing is completed, the device failed its diagnostic, and the status has been set, the device shall clear BSY to zero, negate PGIAG-, and make a transition to the DI2: Device_idle_NS state (see Figure 43).

## 11.2 Software reset protocol

This clause describes the protocol for processing of software reset when the host sets SRST.

If the host sets SRST in the Device Control register to one regardless of the power management mode, the device shall execute the software reset protocol. If the host asserts RESET- before a device has completed the software reset protocol, then the device shall execute the hardware reset protocol from the beginning.

The host should not set the SRST bit to one in the Device Control while the BSY bit is set to one in either device Status register as a result of executing the software reset protocol. If the host sets the SRST bit in the Device Control register to one before devices have completed execution of the software reset protocol, then the devices shall restart execution of the software reset protocol from the beginning. If the host issues a DEVICE RESET command before devices have completed execution of the software reset protocol, the command shall be ignored.

A host should issue an IDENTIFY DEVICE and/or IDENTIFY PACKET DEVICE command after the software reset protocol has completed to determine the current status of features implemented by the device(s).

Figure 38 and the text following the figure decribe the software reset protocol for the host. Figure 39 and the text following the figure describes the software reset protocol for Device 0. Figure 40 and the text following the figure describes the software reset protocol for Device 1.

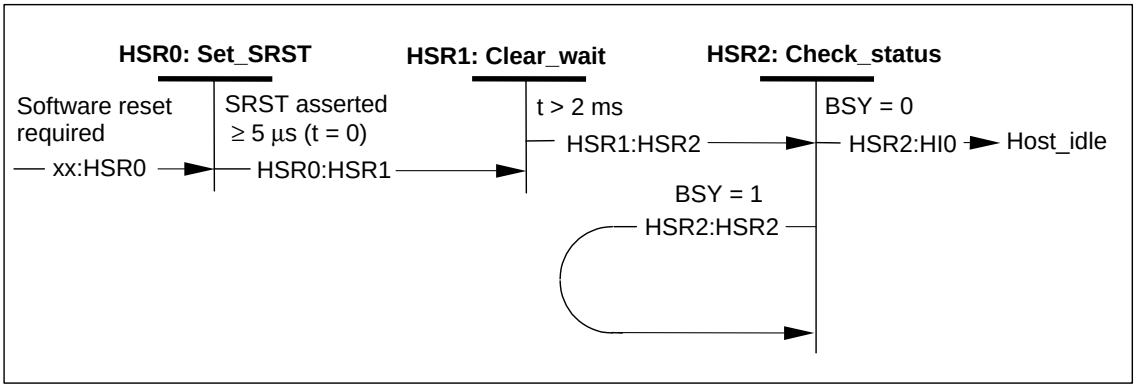



**Figure 38 – Host software reset state diagram**

**HSR0: Set_SRST State:** This state is entered when the host initiates a software reset.

When in this state, the host shall set SRST in the Device Control register to one. The SRST bit shall be written to both devices when the Device Control register is written. The host shall remain in this state with SRST set to one for at least 5 µs. The host shall not set SRST to one unless the bit has been cleared to zero for at least 5 µs.

**Transition HSR0:HSR1:** When the host has had SRST set to one for at least 5 µs, the host shall make a transition to the HSR1: Clear_wait state.

**HSR1: Clear_wait State:** This state is entered when SRST has been set to one for at least 5 µs.

When in this state, the host shall clear SRST in the Device Control register to zero. The host shall remain in this state for at least 2 ms.

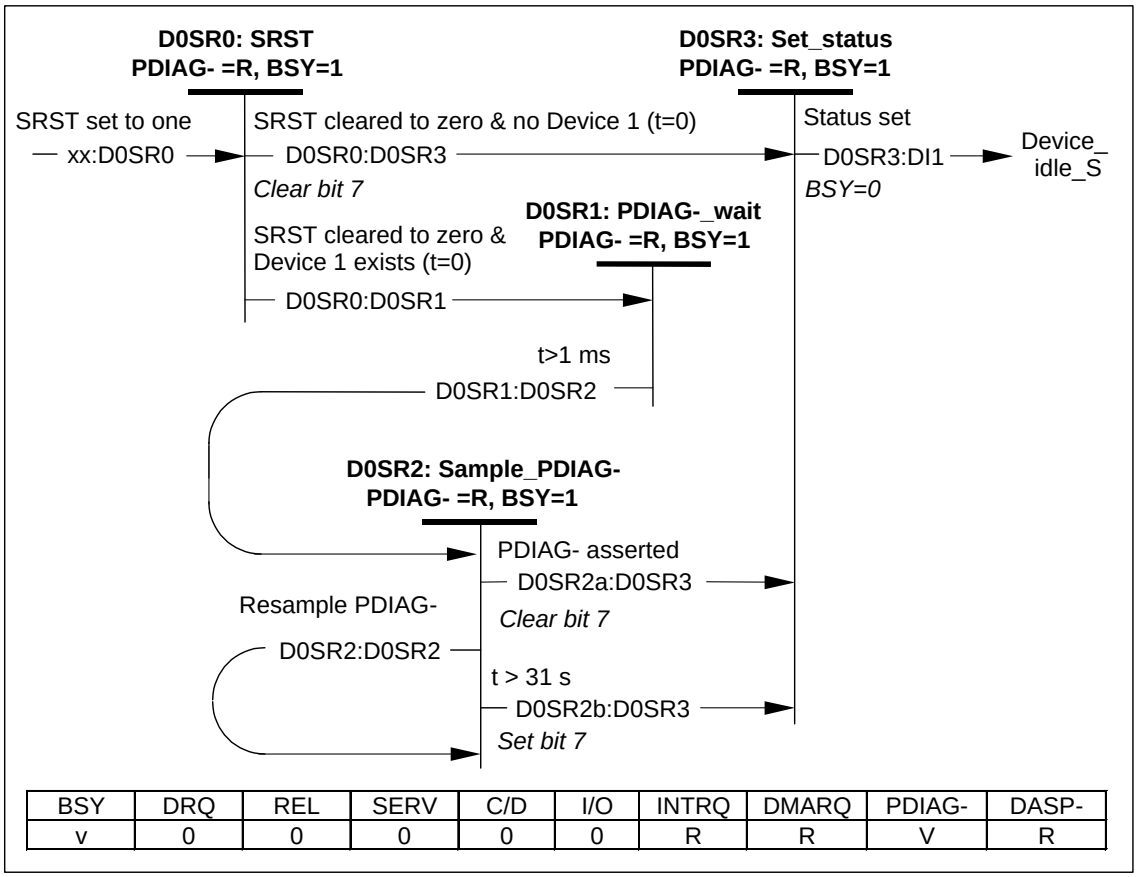
**Transition HSR1:HSR2:** When SRST has been cleared to zero for at least 2 ms, the host shall make a transition to the HSR2: Check_status state.

**HSR2: Check_status State:** This state is entered when SRST has been cleared to zero for at least 2 ms.

When in this state the host shall read the Status or Alternate Status register.

**Transition HSR2:HSR2:** When BSY is set to one, the host shall make a transition to the HSR2: Check_status state.

**Transition HSR2:HI0:** When BSY is cleared to zero, the host shall check the ending status in the Error register and the signature (see clause 5) and make a transition to the HI0: Host_idle state (see Figure 41).




| BSY | DRQ | REL | SERV | C/D | I/O | INTRQ | DMARQ | PDIAG- | DASP- |
|---|---|---|---|---|---|---|---|---|---|
| v | 0 | 0 | 0 | 0 | 0 | R | R | V | R |

**Figure 39 – Device 0 software reset state diagram**

**D0SR0: SRST State:** This state is entered by Device 0 when the SRST bit is set to one in the Device Control register.

When in this state, the device shall release PDIAG-, INTRQ, IORDY, DMARQ, and DD(15:0) within 400 ns after entering this state. The device shall set BSY to one within 400 ns after entering this state.

If the device does not implement the PACKET command feature set, the device should begin performing the hardware initialization and self-diagnostic testing. The device may revert to the default condition (the device's setting may now be in different conditions than they were before the SRST bit was set to one by the host). However, an Ultra DMA mode setting (either enabled or disabled) shall not be affected by the host setting SRST to one.

If the PACKET command feature set is implemented, the device may begin performing the hardware initialization and self-diagnostic testing and the device is not expected to stop any background device activity (e.g., immediate command, see MMC or MMC-2) that was started prior to the time that SRST was set to one. The device shall not revert to the default condition and an Ultra DMA mode setting (either enabled or disabled) shall not be affected by the host setting SRST to one.

**Transition D0SR0:D0SR1:** When SRST is cleared to zero and the assertion of DASP- by Device 1 was detected during the most recent power-on or hardware reset, the device shall make a transition to the D0SR1: PDIAG-_wait state.

**Transition D0SR0:D0SR3:** When SRST is cleared to zero and the assertion of DASP- by Device 1 was not detected during the most recent power-on or hardware reset, the device shall clear bit 7 to zero in the Error register and make a transition to the D0SR3: Set_status state.

**D0SR1: PDIAG-_wait State:** This state is entered when SRST has been cleared to zero and Device 1 is present.

The device shall remain in this state for at least 1 ms and shall clear the DEV bit in the Device register to zero within 1 ms.

**Transition D0SR1:D0SR2:** When at least 1 ms has elapsed since SRST was cleared to zero, the device shall make a transition to the D0SR2: Sample_PDIAG- state.

**D0SR2: Sample_PDIAG- State:** This state is entered when SRST has been cleared to zero for at least 1 ms.

When in this state, the device shall sample the PDIAG- signal.

**Transition D0SR2:D0SR2:** When the sample indicates that PDIAG- is not asserted and less than 31 s have elapsed since SRST was cleared to zero, then the device shall make a transition to the D0SR2: Sample_PDIAG- state.

**Transition D0SR2a:D0SR3:** When the sample indicates that PDIAG- is asserted, the device device shall clear bit 7 to zero in the Error register and shall make a transition to the D0SR3: Set_status state.

**Transition D0SR2b:D0SR3:** When the sample indicates that PDIAG- is not asserted and 31 s have elapsed since SRST was cleared to zero, the device shall set bit 7 to one in the Error register and shall make a transition to the D0SR3: Set_status state.

**D0SR3: Set_status State:** This state is entered when Bit 7 in the Error register has been set or cleared or Device 1 does not exist.

When in this state, the device shall clear the DEV bit in the Device register to zero within 1 ms. The device shall complete any hardware initialization and self-diagnostic testing begun in the SRST state if not already completed.

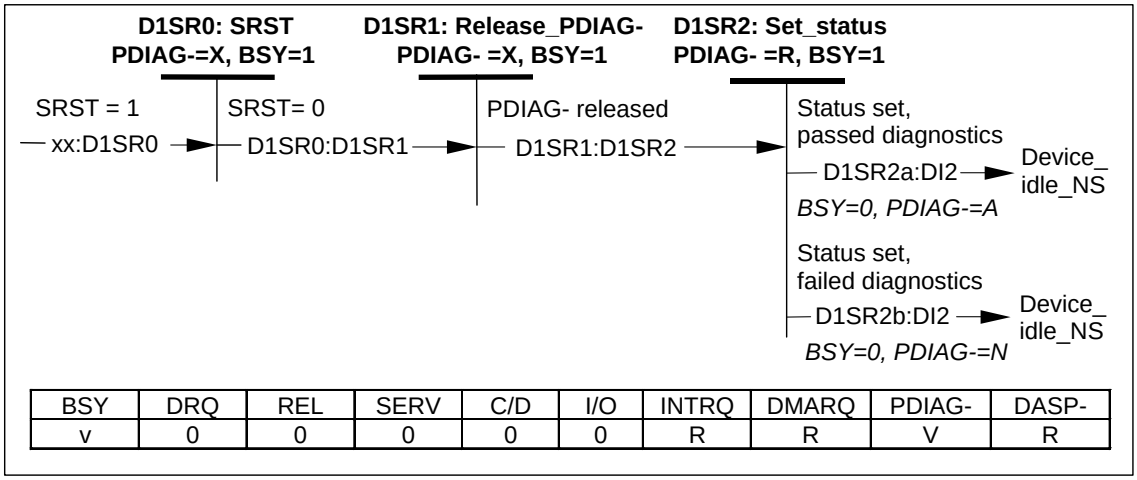
All actions required in this state shall be completed within 31 s.

The EXECUTE DEVICE DIAGNOSTICS diagnostic code shall be placed in bits (6:0) of the Error register (see clause 6). The device shall set the signature values (see clause 5). The content of the Features register is undefined.

If the device does not implement the PACKET command feature set, the device shall clear bits 3, 2, and 0 in the Status register to zero.

If the device implements the PACKET command feature set, the device shall clear bits 6, 5, 4, 3, 2, and 0 in the Status register to zero. The device shall return the operating modes to their specified initial conditions. MODE SELECT conditions shall be restored to their last saved values if saved values have been established. MODE SELECT conditions for which no values have been saved shall be returned to their default values.

**Transition D0SR3:DI1:** When hardware initialization and self-diagnostic testing is completed and the status has been set, the device shall clear BSY to zero and make a transition to the DI1: Device_idle_S state (see Figure 43).




| BSY | DRQ | REL | SERV | C/D | I/O | INTRQ | DMARQ | PDIAG- | DASP- |
|---|---|---|---|---|---|---|---|---|---|
| v | 0 | 0 | 0 | 0 | 0 | R | R | V | R |

**Figure 40** – **Device 1 software reset state diagram**

**D1SR0: SRST State:** This state is entered by Device 1 when the SRST bit is set to one in the Device Control register.

When in this state, the device shall release INTRQ, IORDY, DMARQ, and DD(15:0) within 400 ns after entering this state. The device shall set BSY to one within 400 ns after entering this state.

If the device does not implement the PACKET command feature set, the device shall begin performing the hardware initialization and self-diagnostic testing. The device may revert to the default condition (the device's setting may now be in different conditions than they were before the SRST bit was set to one by the host). However, an Ultra DMA mode setting (either enabled or disabled) shall not be affected by the host setting SRST to one.

If the PACKET command feature set is implemented, the device may begin performing the hardware initialization and self-diagnostic testing and the device is not expected to stop any background device activity (e.g., immediate command, see MMC and MMC-2) that was started prior to the time that SRST was set to one. The device shall not revert to the default condition and an Ultra DMA mode setting (either enabled or disabled) shall not be affected by the host setting SRST to one.

**Transition D1SR0:D1SR1:** When SRST is cleared to zero, the device shall make a transition to the D1SR1: Release_PDIAG- state.

**D1SR1: Release_PDIAG- State:** This state is entered when SRST is cleared to zero.

When in this state, the device shall release PDIAG- and clear the DEV bit in the Device register to zero within 1 ms of entering this state.

**Transition D1SR1:D1SR2:** When PDIAG- has been released, the device shall make a transition to the D1SR2: Set_status state.

**D1SR2: Set_status State:** This state is entered when the device has negated PDIAG-.

When in this state the device shall complete the hardware initialization and self-diagnostic testing begun in the SRST state if not already completed. The EXECUTE DEVICE DIAGNOSTICS diagnostic code shall be placed in the Error register (see clause 6). If the device passed the self-diagnostics, the device shall assert PDIAG-.

All actions required in this state shall be completed within 30 s.

The device shall set the signature values (see clause 5). The contents of the Features register is undefined.

If the device does not implement the PACKET command feature set, the device shall clear bits 3, 2, and 0 in the Status register to zero.

If the device implements the PACKET command feature set, the device shall clear bits 6, 5, 4, 3, 2, and 0 in the Status register to zero. The device shall return the operating modes to their specified initial conditions. MODE SELECT conditions shall be restored to their last saved values if saved values have been established. MODE SELECT conditions for which no values have been saved shall be returned to their default values.

**Transition D1SR2a:DI2:** When hardware initialization, self-diagnostic testing is completed, the device passed the diagnostics, and the status has been set, the device shall clear BSY to zero, assert PDIAG-, and make a transition to the DI2: Device_idle_NS state (see Figure 43).

**Transition D1SR2b:DI2:** When hardware initialization, self-diagnostic testing is completed, the device failed the diagnostics, and the status has been set, the device shall clear BSY to zero, negate PDIAG-, and make a transition to the DI2: Device_idle_NS state (see Figure 43).

## 11.3 Bus idle protocol

When the selected device has BSY cleared to zero and DRQ cleared to zero the bus is idle.

If command overlap is implemented and enabled, the host may be waiting for a service request for a released command. In this case, the device is preparing for the data transfer for the released command.

If command overlap and command queuing are implemented and enabled, the host may be waiting for a service request for a number of released commands. In this case, the device is preparing for the data transfer for one of the released commands.

Figure 41 and the text following the figure describe the host state during bus idle for hosts not implementing command overlap and queuing. Figure 42 and the text following the figure describes the additional host state during bus idle required for command overlap and queuing. Figure 43 and the text following the figure describe the device state during bus idle for devices not implementing command overlap and queuing. Figure 44 and the text following the figure describe the additional device state during bus idle required for command overlap and queuing.

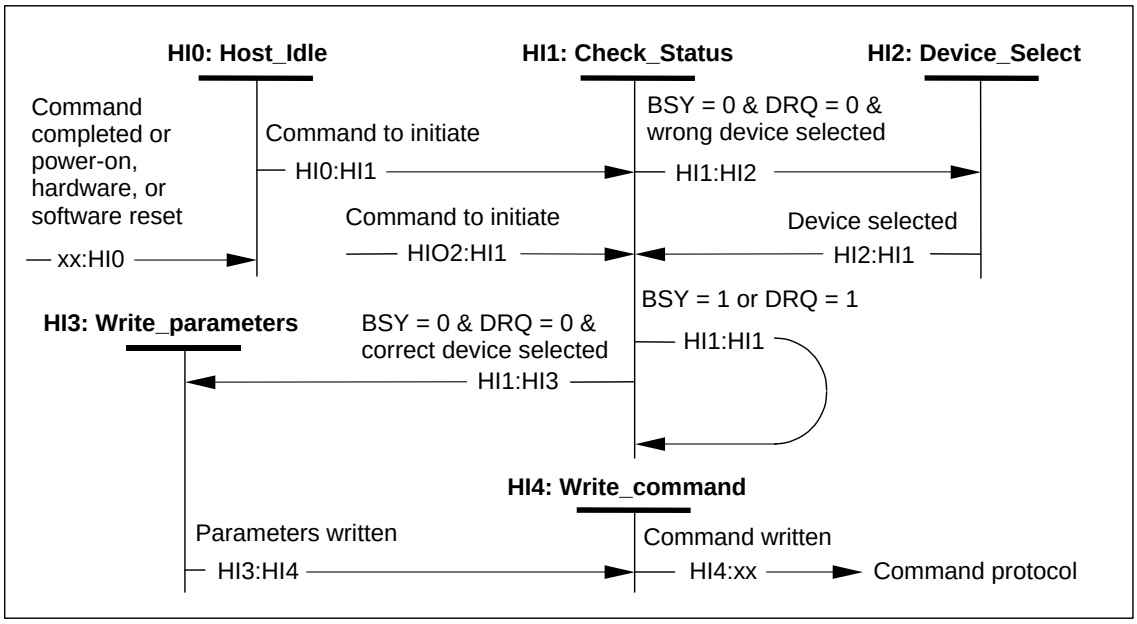



**Figure 41 – Host bus idle state diagram**

**HI0: Host_Idle State:** This state is entered when a device completes a command or when a power-on, hardware, or software reset has occurred.

When in this state, the host waits for a command to be issued to a device.

**Transition HI0:HI1:** When the host has a command to issue to a device, the host shall make a transition to the HI1: Check_Status state.

**HI1: Check_Status State:** This state is entered when the host has a command to issue to a device.

When in this state, the host reads the device Status or Alternate Status register.

**Transition HI1:HI2:** When the status read indicates that both BSY and DRQ are cleared to zero but the wrong device is selected, then the host shall make a transition to the HI2: Device_Select state.

**Transition HI1:HI1:** When the status read indicates that either BSY or DRQ is set to one, the host shall make a transition to the HI1: Check_Status state to recheck the status of the selected device.

**Transition HI1:HI3:** When the status read indicates that both BSY and DRQ are cleared to zero and the correct device is selected, then the host shall make a transition to the HI3: Write_Parameters state.

**HI2: Device_Select State:** This state is entered when the wrong device is selected for issuing a new command.

When in this state, the host shall write to the Device register to select the correct device.

**Transition HI2:HI1:** When the Device register has been written to select the correct device, then the host shall make a transition to the HI1: Check_Status state.

**HI3: Write_Parameters State:** This state is entered when the host has determined that the correct device is selected and both BSY and DRQ are cleared to zero.

When in this state, the host writes all required command parameters to the device Command Block registers (see 6).

**Transition HI3:HI4:** When all required command parameters have been written to the device Command Block registers, the host shall make a transition to the HI4: Write_Command state.

**HI4: Write_Command State:** This state is entered when the host has written all required command parameters to the device Command Block registers.

When in this state, the host writes the command to the device Command register.

**Transition HI4:xx:** When the host has written the command to the device Command register, the host shall make a transition to the command protocol for the command written as described in 11.4 through 11.11.

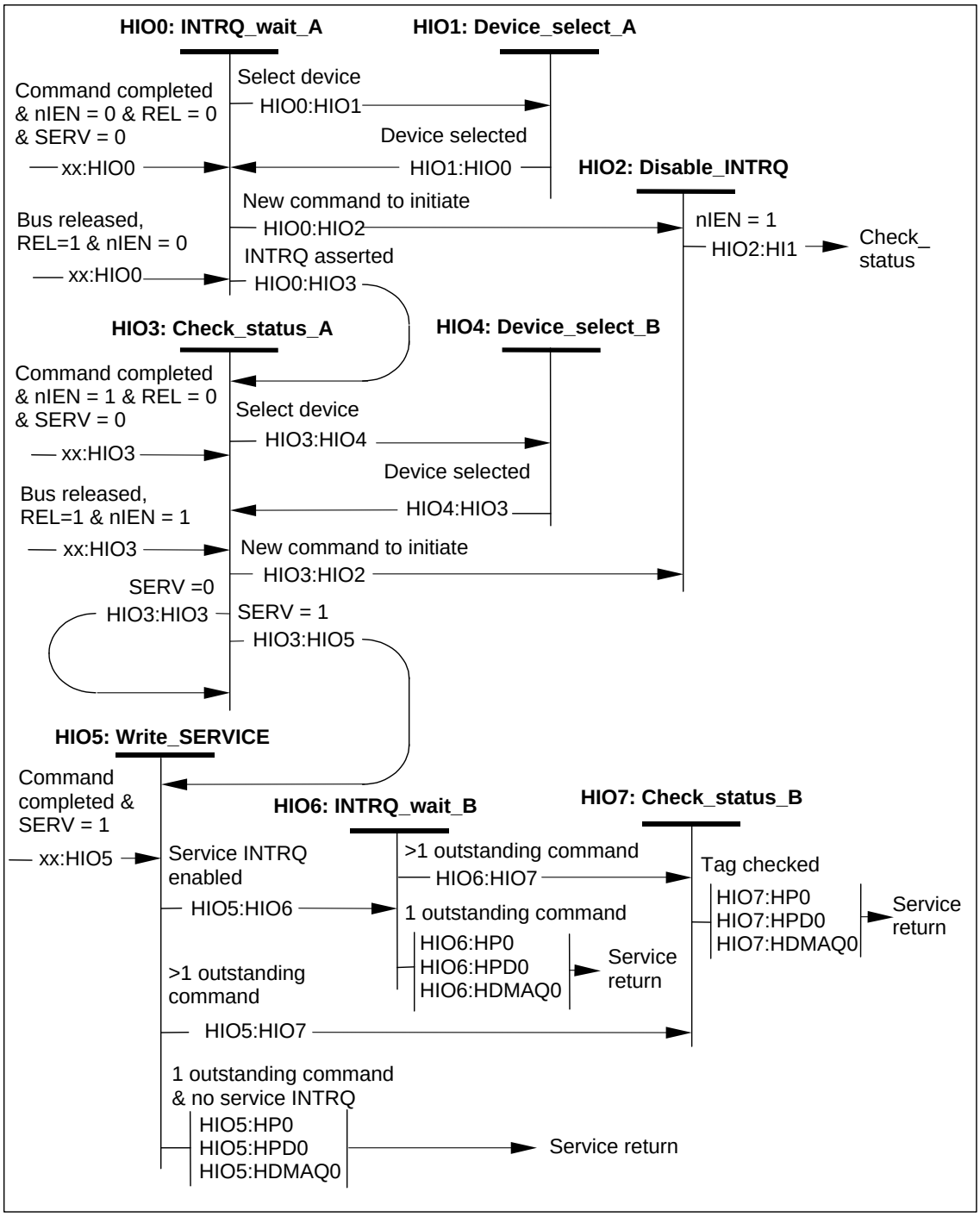



**Figure 42 – Additional Host bus Idle state diagram with Overlap or overlap and queuing**

**HIO0: INTRQ_wait_A State:** This state is entered when a command has completed with nIEN cleared to zero, REL set to one, and SERV cleared to zero. This state is entered when the device has released the bus with nIEN cleared to zero. This state is entered when the host is waiting for INTRQ to be asserted for bus released commands.

When in this state, the host waits for INTRQ to be asserted indicating that a device is ready to resume execution of a bus released command.

**Transition HIO0:HIO1:** When the host has one or more commands outstanding to both devices, the host may make a transition to the HIO1: Device_select_A state to sample INTRQ for the other device.

**Transition HIO0:HIO2:** When the host has a new command to issue to a device and that device has no command released or supports command queuing, then the host shall make a transition to the HIO2: Disable_INTRQ state.

**Transition HIO0:HIO3:** When the host detects INTRQ asserted, the host shall make a transition to the HIO3: Check_status A state.

**HIO1: Device_select_A State:** This state is entered when the host has outstanding, bus released commands to both devices and nIEN is cleared to zero.

When in this state, the host shall disable INTRQ by setting nIEN to one, shall write the Device register to select the other device, and then, shall enable INTRQ by clearing nIEN to zero.

**Transition HIO1:HIO0:** Having selected the other device, the host shall make a transition to the HIO0: INTRQ_wait_A state.

**HIO2: Disable_INTRQ State:** This state is entered when the host has a new command to issue to a device and that device has no outstanding, bus released command or supports command queuing.

When in this state, the host shall set nIEN to one. nIEN is set to one to prevent a race condition if the host has to select the other device to issue the command.

**Transition HIO2:HI1:** When nIEN has been set to one, the host shall make a transition to the HI1: Check_status state (see Figure 41).

**HIO3: Check_status_A State:** This state is entered when a command is completed with nIEN set to one, REL set to one, and SERV cleared to zero. This state is entered when the device has released the bus and nIEN is set to one. This state is entered when an interrupt has occured indicating that a device is requesting service.

When in this state, the host shall read the Status register of the device requesting service.

**Transition HIO3:HIO4:** If SERV is cleared to zero and the host has released commands outstanding to both devices, then the host may make a transition to the HIO4: Device_select_B state.

**Transition HIO3:HIO2:** If SERV is cleared to zero and the host has a new command to issue to a device, then the host shall make a transition to the HIO2: Disable_INTRQ state.

**Transition HIO3:HIO3:** If SERV is cleared to zero and the host has no new command to issue, then the host shall make a transition to the HIO3: Check_status state.

**Transition HIO3:HIO5:** If SERV is set to one, the host shall make a transition to the HIO5: Write_SERVICE state.

**HIO4: Device_select_B State:** This state is entered when the host has outstanding, bus released commands to both devices and nIEN is set to one.

When in this state, the host shall disable INTRQ by setting nIEN to one, shall write the Device register to select the other device, and then, shall enable INTRQ by clearing nIEN to zero.

**Transition HIO4:HIO3:** Having selected the other device, the host shall make a transition to the HIO3: Check_status_A state.

**HIO5: Write_SERVICE State:** This state is entered when a device has set SERV to one indicating that the device requests service. This state is entered when a command has completed with SERV set to one.

When in this state, the host shall write the SERVICE command to the Command register.

**Transition HIO5:HIO6:** When the device is one that implements the PACKET command feature set and the Service interrupt is enabled, then the host shall make a transition to the HIO6: INTRQ_wait_B state.

**Transition HIO5:HIO7:** When the host has more than one released command outstanding to the device and the Service interrupt is disabled, the host shall make a transition to the HIO7: Check_status_B state.

**Transition HIO5:xx:** When the Service interrupt is disabled and the host has only one released command outstanding to the device, the host shall make a transition to the service return for the protocol for the command outstanding (see Figure 53, Figure 55, or Figure 57).

**HIO6: INTRQ_wait_B State:** This state is entered when the SERVICE command has been written to a device implementing the PACKET command feature set and the Service interrupt is enabled.

> NOTE – READ DMA QUEUED and WRITE DMA QUEUED commands do not implement the Service interrupt.

When in this state, the host waits for the assertion of INTRQ.

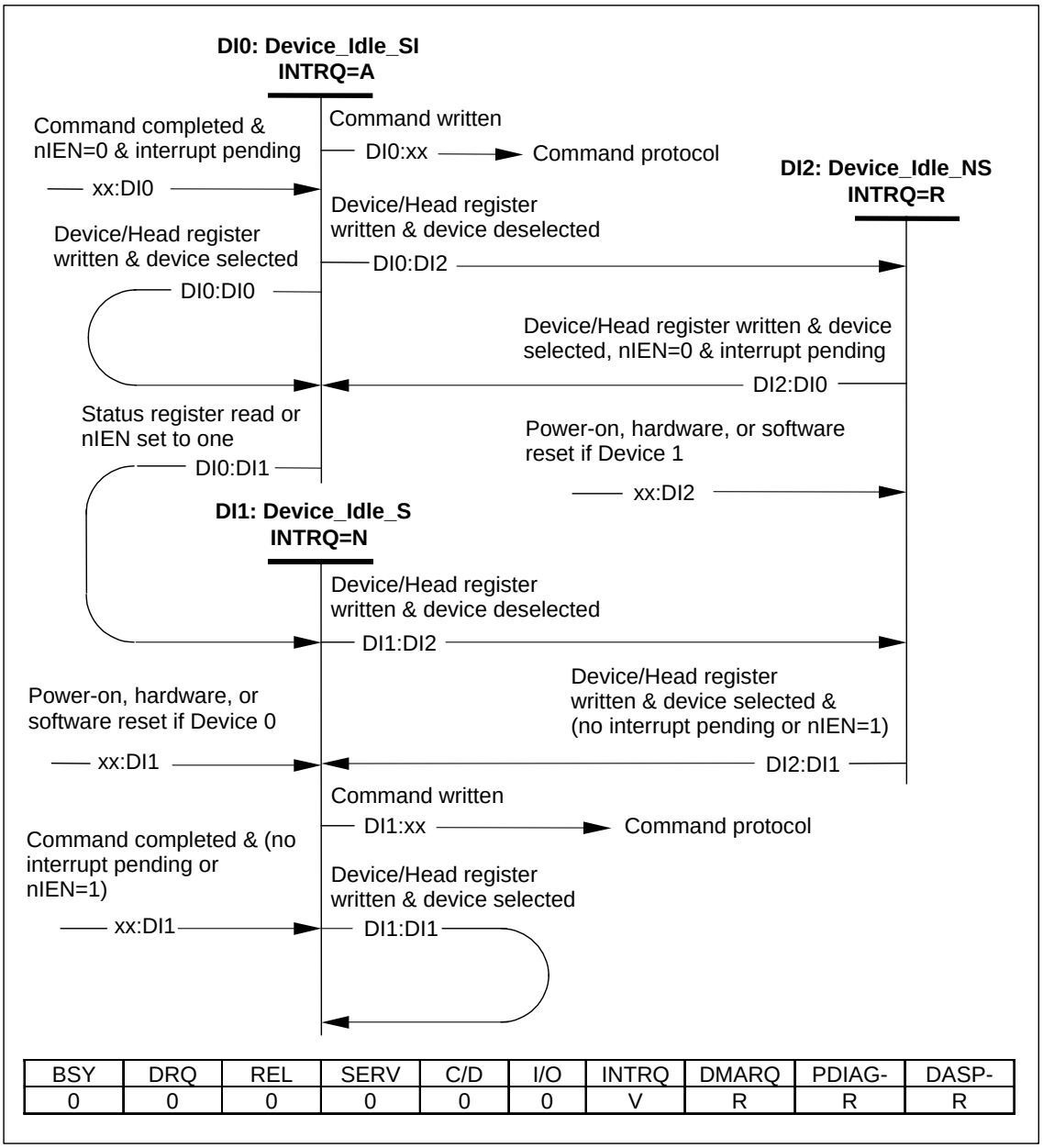
**Transition HIO6:HIO7:** When the host has more than one released command outstanding to the device and INTRQ is asserted, the host shall make a transition to the HIO7: Check_status_B state.

**Transition HIO6:xx:** When INTRQ has been asserted and the host has only one released command outstanding to the device, then the host shall make a transition to the service return for the protocol for the command outstanding (see Figure 53, Figure 55, or Figure 57).

**HIO7: Check_status_B State:** This state is entered when the SERVICE command has been written and the host has more than one released command outstanding to the device.

When in this state the host reads the command tag to determine which outstanding command service is requested for. If a DMA data transfer is required for the command, the host shall set up the DMA engine.

**Transition HIO7:xx:** When the command for which service is requested has been determined, the host shall make a transition to the service return for that command protocol (see Figure 53, Figure 55, or Figure 57).




| BSY | DRQ | REL | SERV | C/D | I/O | INTRQ | DMARQ | PDIAG- | DASP- |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | V | R | R | R |

**Figure 43** – **Device bus Idle state diagram**

**DI0: Device_Idle_SI State** (selected/INTRQ asserted)**:** This state is entered when the device has completed the execution of a command protocol with interrupt pending and nIEN=0.

When in this state, the device shall have DRQ cleared to zero, INTRQ asserted, and BSY cleared to zero. Reading any register except the Status register shall have no effect.

**Transition DI0:xx:** If the Command register is written, the device shall clear the device internal interrupt pending, shall negate or release INTRQ within 400 ns of the negation of DIOW-, shall release PDIAG- and DSAP- if asserted, and shall make a transition to the command protocol indicated by the content of the Command register. The host should not write to the Command register at this time.

**Transition DI0:DI1:** When the Status register is read, the device shall clear the device internal interrupt pending, negate or release INTRQ within 400 ns of the negation of DIOR-, and make a transition to the DI1: Device_Idle_S state. When nIEN is set to one in the Device Control register, the device shall negate INTRQ and make a transition to the DI1: Device_Idle_S state.

**Transition DI0:DI0:** When the Device register is written and the DEV bit selects this device or any other register except the Command register is written, the device shall make a transition to the DI0: Device_Idle_SI state.

**Transition DI0:DI2:** When the Device register is written and the DEV bit selects the other device, then the device shall release INTRQ within 400 ns of the negation of DIOW-, and make a transition to the DI2: Device_Idle_NS state.

**DI1: Device_Idle_S State** (selected/INTRQ negated)**:** This state is entered when the device has completed the execution of a command protocol with no interrupt pending or nIEN=1, or when a pending interrupt is cleared. This state is also entered by Device 0 at the completion of a power-on, hardware, or software reset.

When in this state, the device shall have BSY and DRQ cleared to zero and INTRQ negated or released.

When entering this state from a power-on, hardware, or software reset, if the device does not implement the PACKET command feature set, the device shall set DRDY to one within 30 s of entering this state. When entering this state from a power-on, hardware, or software reset, if the device does implement the PACKET command feature set, the device shall not set DRDY to one.

**Transition DI1:xx:** When the Command register is written, the device shall exit the interrupt pending state, release PDIAG- if asserted and make a transition to the command protocol indicated by the content of the Command register.

**Transition DI1:DI1:** When the Device register is written and the DEV bit selects this device or any register is written except the Command register, the device shall make a transition to the DI1: Device_Idle_S state.

**Transition DI1:DI2:** When the Device register is written and the DEV bit selects the other device, the device shall make a transition to the DI2: Device_Idle_NS state.

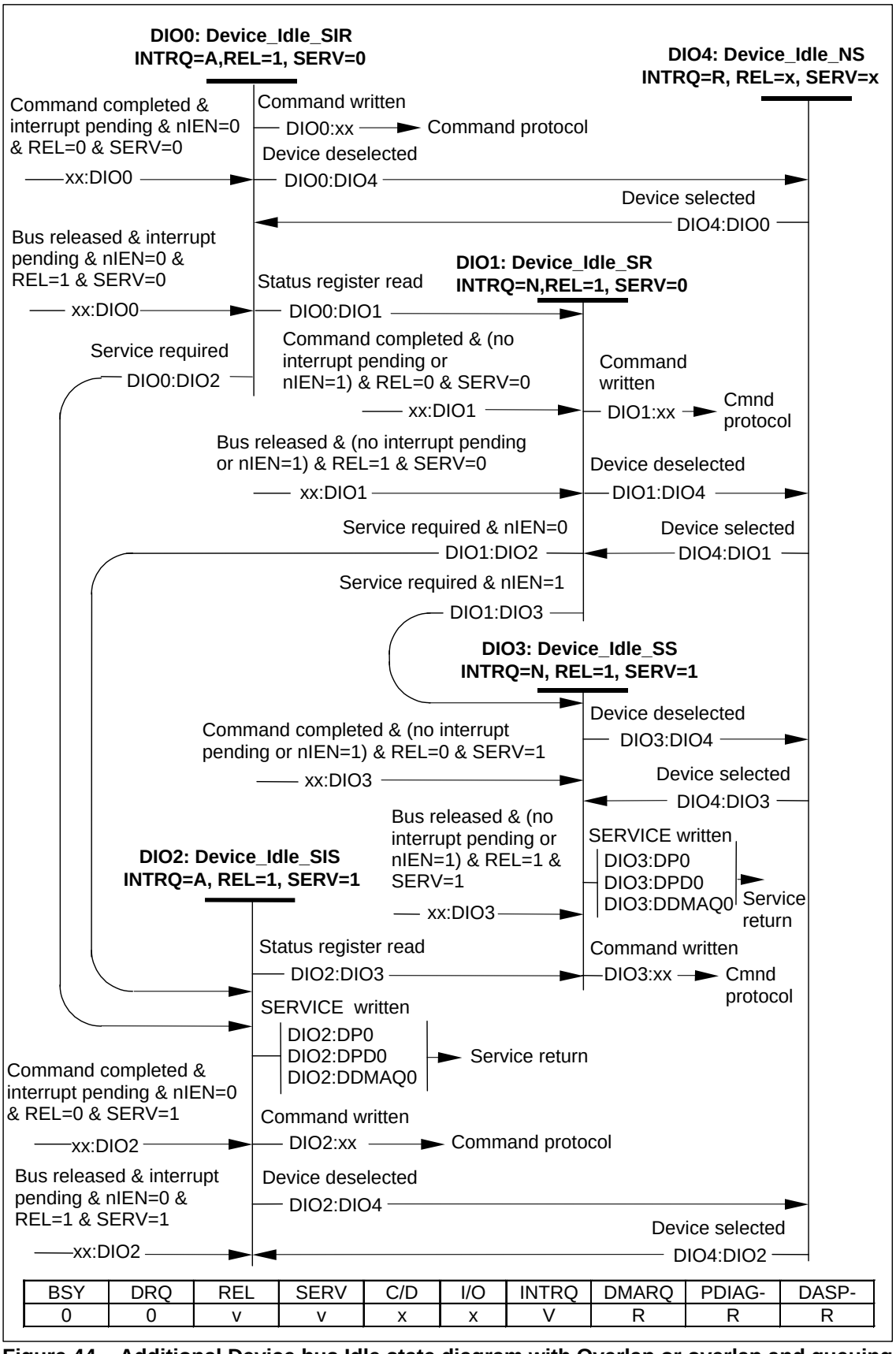
**DI2: Device_Idle_NS State** (not selected)**:** This state is entered when the device is deselected. This state is also entered by Device 1 at the completion of a power-on, hardware, or software reset.

When in this state, the device shall have BSY and DRQ cleared to zero and INTRQ shall be released.

When entering this state from a power-on, hardware, or software reset, if the device does not implement the PACKET command feature set, the device shall set DRDY to one within 30 s of entering this state and shall release DASP- and PDIAG- with 31 s of entering this state. When entering this state from a power-on, hardware, or software reset, if the device does implement the PACKET command feature set, the device shall not set DRDY to one.

**Transition DI2:DI0:** When the Device register is written, the DEV bit selects this device, the device has an interrupt pending, and nIEN is cleared to zero, then the device shall assert INTRQ within 400 ns of the negation of DIOW- and make a transition to the DI0: Device_Idle_SI state.

**Transition DI2:DI1:** When the Device register is written, the DEV bit selects this device, and the device has no interrupt pending or nIEN is set to one, then the device shall make a transition to the DI1: Device_Idle_S state.




| BSY | DRQ | REL | SERV | C/D | I/O | INTRQ | DMARQ | PDIAG- | DASP- |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | v | v | x | x | V | R | R | R |

**Figure 44 – Additional Device bus Idle state diagram with Overlap or overlap and queuing**

**DIO0: Device_Idle_SIR State** (selected/INTRQ asserted/RELset to one)**:** This state is entered when the device has completed the execution of a command protocol with interrupt pending, nIEN=0, REL set to one, and SERV cleared to zero. This state is entered when the device has released an overlapped command with interrupt pending, nIEN=0, REL set to one, and SERV cleared to zero.

When in this state, the device is preparing for completion of a released command. The device shall have BSY and DRQ cleared to zero, and INTRQ asserted.

**Transition DIO0:xx:** When the Command register is written, the device shall clear the interrupt pending, shall negate or release INTRQ within 400 ns of the negation of DIOW-, and shall make a transition to the command protocol indicated by the content of the Command register.

> NOTE – Since a queue exists, only commands in the queued command set may be written to the Command register. If any other command is written to the Command register, the queue is aborted and command aborted is returned for the command (see **Error! Reference source not found.**).

**Transition DIO0:DIO1:** When the Status register is read, the device shall clear the interrupt pending, negate or release INTRQ within 400 ns of the negation of DIOR-, and make a transition to the DIO1: Device_Idle_SR state.

**Transition DIO0:DIO2:** When the Device register is written and the DEV bit selects the other device, then the device shall release INTRQ within 400 ns of the negation of DIOW- and make a transition to the DIO2: Device_Idle_NS state.

**Transition DIO0:DIO2:** When the device is ready to continue the execution of a released command, the device shall make a transition to the DIO2: Device_idle_SIS state.

**DIO1: Device_Idle_SR State** (selected/INTRQ negated/REL set to one)**:** This state is entered when the device has completed the execution of a command protocol with no interrupt pending or nIEN=1, REL set to one, and SERV cleared to zero. This state is entered when the device has released an overlapped command with no interrupt pending or nIEN=1, REL set to one, and SERV cleared to zero. This state is entered when a pending interrupt is cleared, REL is set to one, and SERV is cleared to zero.

When in this state, the device is preparing for completion of a released command. The device shall have BSY and DRQ cleared to zero, and INTRQ negated or released.

**Transition DIO1:xx:** When the Command register is written, the device shall make a transition to the command protocol indicated by the content of the Command register.

> NOTE – Since a queue exists, only commands in the queued command set may be written to the Command register. If any other command is written to the Command register, the queue is aborted and command aborted is returned for the command (see **Error! Reference source not found.**).

**Transition DIO1:DIO4:** When the Device register is written and the DEV bit selects the other device, the device shall make a transition to the DIO4: Device_Idle_NS state.

**Transition DIO1:DIO2:** When the device is ready to continue the execution of a released command and nIEN=0, the device shall make a transition to the DIO2: Device_idle_SIS state.

**Transition DIO1:DIO3:** When the device is ready to continue the execution of a released command and nIEN=1, the device shall make a transition to the DIO3: Device_idle_SS state.

**DIO2: Device_Idle_SIS State** (selected/INTRQ asserted/SERV set to one)**:** This state is entered when the device has completed the execution of a command protocol with interrupt pending,

nIEN=0, REL set to one, and SERV set to one. This state is entered when the device has released an overlapped with interrupt pending, nIEN=0, REL set to one, and SERV set to one.

**Transition DIO2:DIO3:** When the Status register is read, the device shall clear the interrupt pending, negate or release INTRQ within 400 ns of the negation of DIOR-, and make a transition to the DIO3: Device_Idle_SS state.

**Transition DIO2: DIO4:** When the Device register is written and the DEV bit selects the other device, the device shall release INTRQ within 400 ns of the negation of DIOW- and make a transition to the DIO4: Device_Idle_NS state.

**Transition DIO2:DP0/DPD0/DDMAQ0:** When the SERVICE command is written into the Command register, the device shall set the Tag for the command to be serviced, negate or release INTRQ within 400 ns of the negation of DIOW-, and make a transition to the Service return of the command ready for service (see Figure 54 Device PACKET non-data and PIO data command protocol, Figure 56 Device PACKET DMA command protocol, or Figure 58 Device DMA QUEUED command protocol).

**Transition DIO2:xx:** When any overlapped command other than SERVICE is written to the Command register, the device shall negate or release INTRQ within 400 ns of the negation of DIOW- and make a transition to the protocol for the new command.

**DIO3: Device_Idle_SS State** (selected/INTRQ negated/SERV set to one)**:** This state is entered when the device has completed the execution of a command protocol with no interrupt pending or nIEN=1, REL set to one, and SERV set to one. This state is entered when the device has released an overlapped with no interrupt pending or nIEN=1, REL set to one, and SERV set to one.

**Transition DIO3: DIO4:** When the Device register is written and the DEV bit selects the other device, the device shall make a transition to the DIO4: Device_Idle_NS state.

**Transition DIO3:DP0/DPD0/DDMAQ0:** When the SERVICE command is written into the Command register, the device shall set the Tag for the command to be serviced and make a transition to the Service return of the command ready for service (see Figure 54, Figure 56, or Figure 58).

**Transition DIO3:xx:** When any overlapped command other than SERVICE is written to the Command register, the device shall make a transition to the protocol for the new command.

**DIO4: Device_Idle_NS State** (not selected)**:** This state is entered when the device is deselected with REL or SERV set to one.

When in this state, the device shall have BSY and DRQ cleared to zero and INTRQ shall be released.

**Transition DIO4:DIO0:** When the Device register is written, the DEV bit selects this device, the device has an interrupt pending, nIEN is cleared to zero, REL is set to one, and SERV is cleared to zero, then the device shall assert INTRQ within 400 ns of the negation of DIOW- and make a transition to the DIO0: Device_Idle_SIR state.

**Transition DIO4:DIO1:** When the Device register is written, the DEV bit selects this device, the device has no interrupt pending or nIEN is set to one, REL is set to one, and SERV is cleared to zero, then the device shall make a transition to the DIO1: Device_Idle_SIR state.

**Transition DIO4:DIO2:** When the Device register is written, the DEV bit selects this device, the device has an interrupt pending, nIEN is cleared to zero, REL is set to one, and SERV is set to one, then the device shall assert INTRQ within 400 ns of the negation of DIOW- and make a transition to the DIO2: Device_Idle_SIS state.

**Transition DIO4:DIO3:** When the Device register is written, the DEV bit selects this device, the device has no interrupt pending or nIEN is set to one, REL is set to one, and SERV is set to one, then the device shall make a transition to the DIO3: Device_Idle_SIR state.

## 11.4 Non-data command protocol

This class includes:

- CFA ERASE SECTORS
- CFA REQUEST EXTENDED ERROR CODE
- CHECK MEDIA CARD TYPE
- CHECK POWER MODE
- CONFIGURE STREAM
- DEVICE CONFIGURATION FREEZE LOCK
- DEVICE CONFIGURATION RESTORE
- FLUSH CACHE
- FLUSH CACHE EXT
- GET MEDIA STATUS
- IDLE
- IDLE IMMEDIATE
- MEDIA EJECT
- MEDIA LOCK
- MEDIA UNLOCK
- NOP
- READ NATIVE MAX ADDRESS
- READ NATIVE MAX ADDRESS EXT
- READ VERIFY SECTOR(S)
- READ VERIFY SECTOR(S) EXT
- SECURITY ERASE PREPARE
- SECURITY FREEZE LOCK
- SET FEATURES
- SET MAX ADDRESS
- SET MAX ADDRESS EXT
- SET MULTIPLE MODE
- SLEEP
- SMART DISABLE OPERATION
- SMART ENABLE/DISABLE AUTOSAVE
- SMART ENABLE OPERATIONS
- SMART EXECUTE OFFLINE IMMEDIATE
- SMART RETURN STATUS
- STANDBY
- STANDBY IMMEDIATE

Execution of these commands involves no data transfer. Figure 45 and the text following the figure describes the host state. Figure 46 and the text following the figure decribes the device state.

See the NOP command description and the SLEEP command in clause 6 for additional protocol requirements.

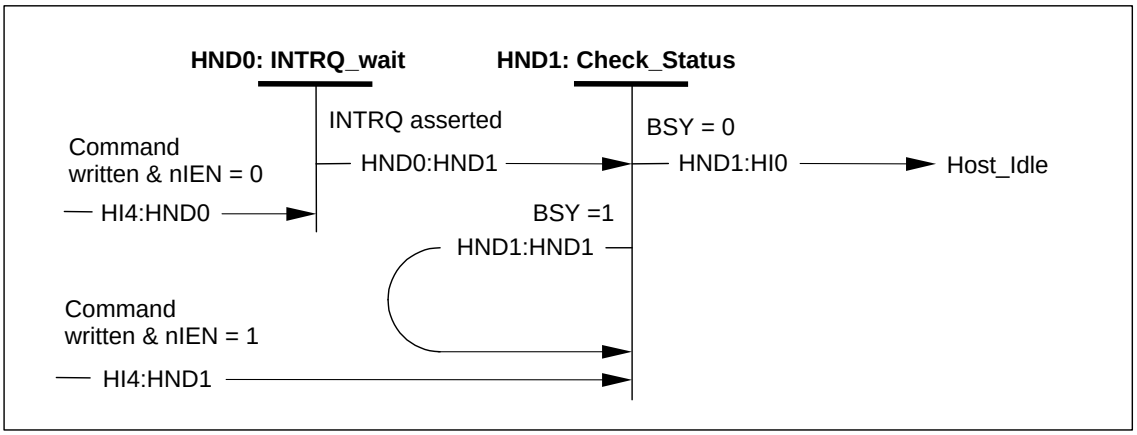



**Figure 45 – Host Non-Data state diagram**

**HND0: INTRQ_Wait_State:** This state is entered when the host has written a non-data command to the device and the nIEN bit in the device has been cleared to zero.

When in this state the host may wait for INTRQ to be asserted by the device.

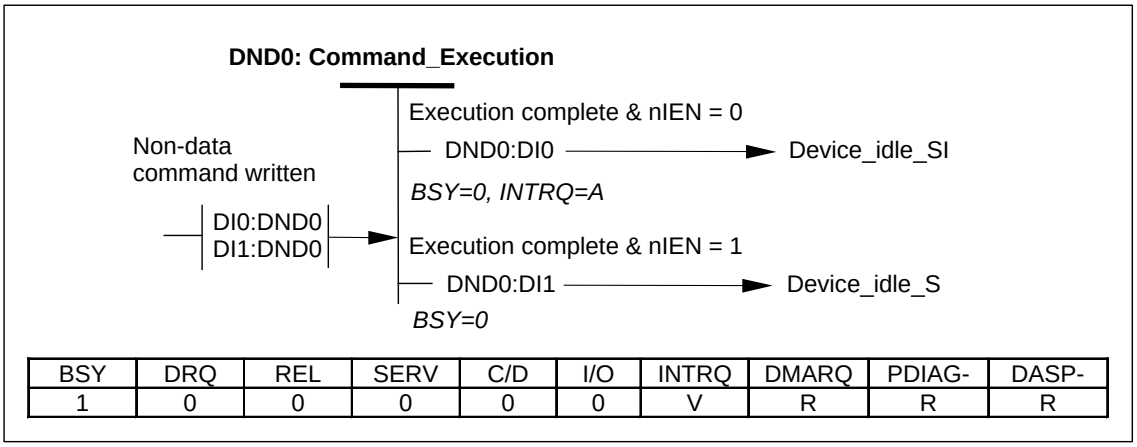
**Transition HND0:HND1:** When the device asserts INTRQ, the host shall make a transition to the HND1: Check_Status state.

**HND1: Check_Status State:** This state is entered when the host has written a non-data command to the device and the nIEN bit in the device has been set to one, or when INTRQ has been asserted.

When in this state, the host shall read the device Status register. When entering this state from another state other than when an interrupt has occurred, the host shall wait 400 ns before reading the Status register.

**Transition HND1:HI0:** When the status read indicates that BSY is cleared to zero, the host shall make a transition to the HI0: Host_Idle state (see Figure 41). If status indicates that an error has occured, the host shall take appropriate error recovery action.

**Transition HND1:HND1:** When the status read indicates that BSY is set to one, the host shall make a transition to the HND1: Check_Status state to recheck device status.




| BSY | DRQ | REL | SERV | C/D | I/O | INTRQ | DMARQ | PDIAG- | DASP- |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | V | R | R | R |

**Figure 46 – Device Non-Data state diagram**

**DND0: Command_Execution State:** This state is entered when a non-data command has been written to the device Command register.

When in this state, the device shall set BSY to one within 400 ns of the writing of the Command register, shall execute the requested command, and shall set the interrupt pending.

**Transition DND0:DI0:** When command execution completes and nIEN is cleared to zero, then the device shall set error bits if appropriate, clear BSY to zero, assert INTRQ, and make a transition to the DI0: Device_Idle_SI state (see Figure 43).

**Transition DND0:DI1:** When command execution completes and nIEN is set to one, the device shall set error bits if appropriate, clear BSY to zero, and make a transition to the DI1: Device_Idle_S state (see Figure 43).

## 11.5 PIO data-in command protocol

This class includes:

- CFA TRANSLATE SECTOR
- DEVICE CONFIGURATION IDENTIFY
- IDENTIFY DEVICE
- IDENTIFY PACKET DEVICE
- READ BUFFER
- READ LOG EXT
- READ MULTIPLE
- READ MULTIPLE EXT
- READ SECTOR(S)
- READ SECTOR(S) EXT
- SMART READ DATA
- SMART READ LOG
- READ STREAM PIO EXT

Execution of this class of command includes the transfer of one or more blocks of data from the device to the host. Figure 47 and the text following the figure describes the host states. Figure 48 and the text following the figure describes the device states.

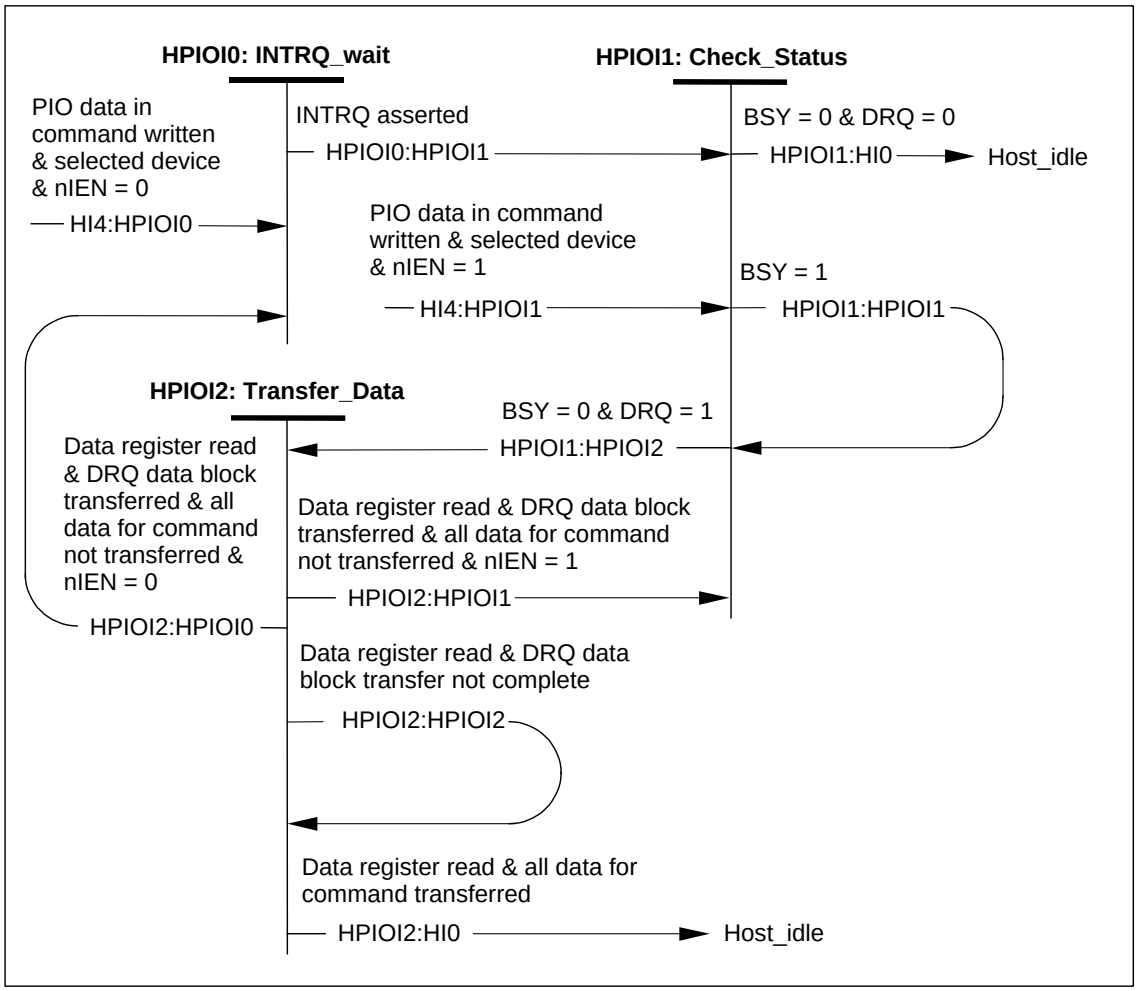



**Figure 47 – Host PIO data-In state diagram**

**HPIOI0: INTRQ_Wait State:** This state is entered when the host has written a PIO data-in command to the device and nIEN is cleared to zero, or at the completion of a DRQ data block transfer if all the data for the command has not been transferred and nIEN is cleared to zero.

When in this state, the host shall wait for INTRQ to be asserted.

**Transition HPIOI0:HPIOI1:** When INTRQ is asserted, the host shall make a transition to the HPIOI1: Check_Status state.

**HPIOI1: Check_Status State:** This state is entered when the host has written a PIO data-in command to the device and nIEN is set to one, or when INTRQ is asserted.

When in this state, the host shall read the device Status register. When entering this state from the HI4 state, the host shall wait 400 ns before reading the Status register. When entering this state from the HPIOI2 state, the host shall wait one PIO transfer cycle time before reading the Status register. The wait may be accomplished by reading the Alternate Status register and ignoring the result.

**Transition HPIOI1:HI0:** When BSY is cleared to zero and DRQ is cleared to zero, then the device has completed the command with an error. The host shall perform appropriate error recovery and make a transition to the HI0: Host_Idle state (see Figure 41).

**Transition HPIOI1:HPIOI1:** When BSY is set to one , the host shall make a transition to the HPIOI1: Check_Status state.

**Transition HPIOI1:HPIOI2:** When BSY is cleared to zero and DRQ is set to one, the host shall make a transition to the HPIOI2: Transfer_Data state.

**HPIOI2: Transfer_Data State:** This state is entered when the BSY is cleared to zero, DRQ is set to one, and the DRQ data block transfer has not completed.

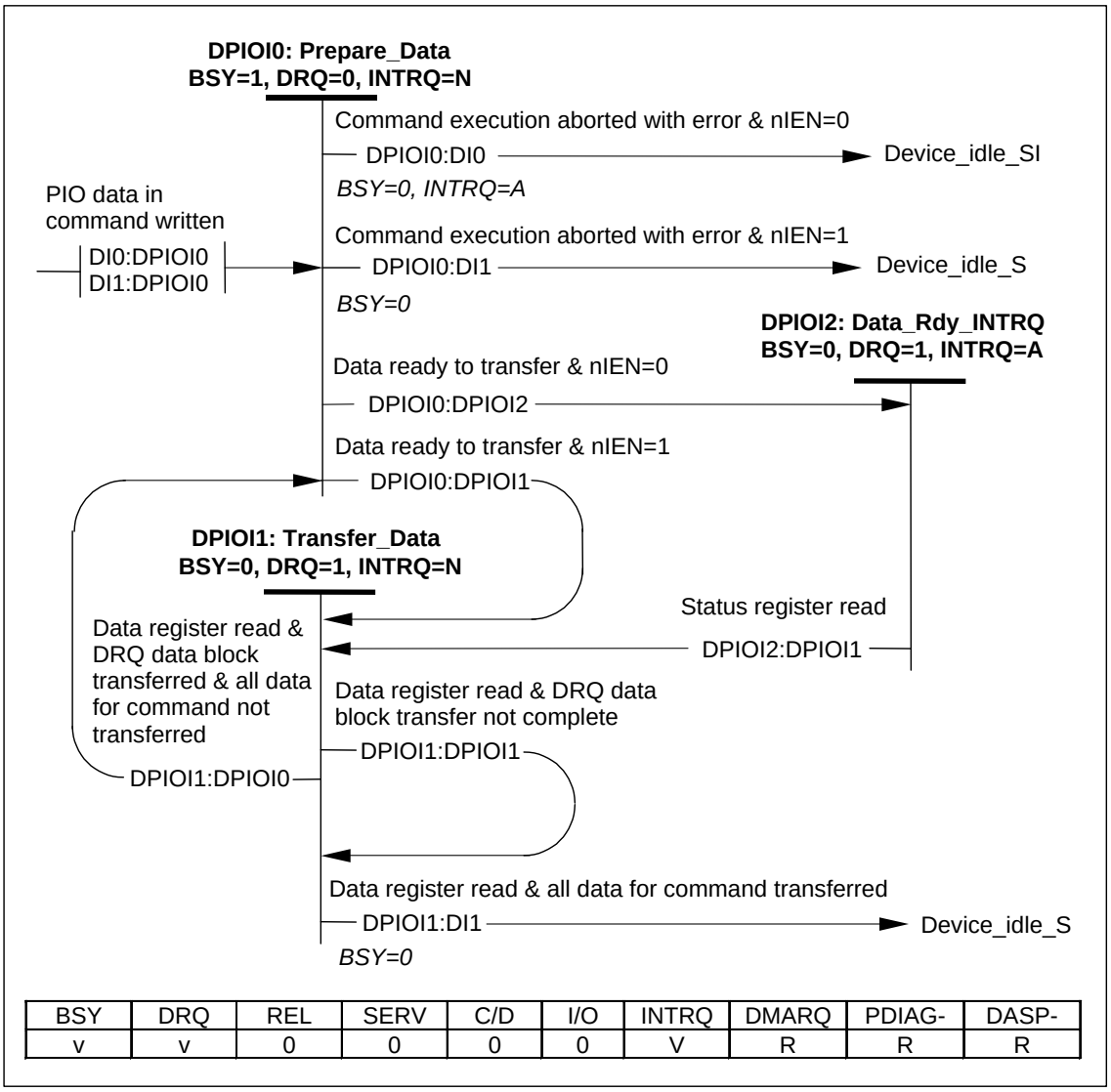
When in this state, the host shall read the device Data register to transfer data.

**Transition HPIOI2:HPIOI0:** When the host has read the device Data register and the DRQ data block has been transferred, all blocks for the command have not been transferred, and nIEN is cleared to zero, then the host shall make a transition to the HPIOI0: INTRQ_Wait state.

**Transition HPIOI2:HPIOI1:** When the host has read the device Data register and the DRQ data block has been transferred, all blocks for the command have not been transferred, and nIEN is set to one, then the host shall make a transition to the HPIOI1: Check_Status state.

**Transition HPIOI2:HPIOI2:** When the host has read the device status register and the DRQ data block transfer has not completed, then the host shall make a transition to the HPIOI2: Transfer_Data state.

**Transition HPIOI2:HI0:** When the host has read the device Data register and all blocks for the command have been transferred, then the host shall make a transition to the HI0: Host_Idle state (see Figure 41). The host may read the Status register.




| BSY | DRQ | REL | SERV | C/D | I/O | INTRQ | DMARQ | PDIAG- | DASP- |
|---|---|---|---|---|---|---|---|---|---|
| v | v | 0 | 0 | 0 | 0 | V | R | R | R |

**Figure 48 – Device PIO data-In state diagram**

**DPIOI0: Prepare_Data State:** This state is entered when the device has a PIO data-in command written to the Command register.

When in this state, device shall set BSY to one within 400 ns of the writing of the Command register and prepare the requested data for transfer to the host.

For IDENTIFY DEVICE and IDENTIFY PACKET DEVICE commands, if the device tests CBLID- it shall do so and update bit 13 in word 93.

**Transition DPIOI0:DI0:** When an error is detected that causes the command to abort and nIEN is cleared to zero, then the device shall set the appropriate error bits, clear BSY to zero, assert INTRQ, and make a transition to the DI0: Device_Idle_SI state (see Figure 43).

**Transition DPIOI0:DI1:** When an error is detected that causes the command to abort and nIEN is set to one, then the device shall set the appropriate error bits, clear BSY to zero, and make a transition to the DI1: Device_Idle_S state (see Figure 43).

**Transition DPIOI0:DPIOI1:** When the device has a DRQ data block ready to transfer and nIEN is set to one, then the device shall make a transition to the DPIOI1: Transfer_Data state.

**Transition DPIOI0:DPIOI2:** When the device has a DRQ data block ready to transfer and nIEN is cleared to zero, then the device shall make a transition to the DPIOI2: Data_Ready_INTRQ state.

**DPIOI1: Data_Transfer State:** This state is entered when the device is ready to transfer a DRQ data block and nIEN is set to one, or when the INTRQ indicating that the device is ready to transfer a DRQ data block has been acknowleged by a read of the Status register.

When in this state, BSY is cleared to zero, DRQ is set to one, INTRQ is negated, and the device has a data word ready in the Data register for transfer to the host.

**Transition DPIOI1:DPIOI1:** When the Data register is read and transfer of the DRQ data block has not completed, then the device shall make a transition to the DPIOI1: Data_Transfer state.

**Transition DPIOI1:DPIOI0:** When the Data register is read and the transfer of the current DRQ data block has completed, but all blocks for this request have not been transferred, then the device shall make a transition to the DPIOI0: Prepare_Data state.

**Transition DPIOI1:DI1:** When the Data register is read and all blocks for this request have been transferred, then the device shall clear BSY to zero and make a transition to the DI1: Device_Idle_S state (see Figure 43). The interrupt pending is not set on this transition.

**DPIOI2: Data_Ready_INTRQ State:** This state is entered when the device has a DRQ data block ready to transfer and nIEN is cleared to zero.

When in this state, BSY is cleared to zero, DRQ is set to one, and INTRQ is asserted.

**Transition DPIOI2:DPIOI1:** When the Status register is read, then the device shall clear the interrupt pending, negate INTRQ, and make a transition to the DPIOI1: Data_Transfer state.

## 11.6 PIO data-out command protocol

This class includes:

- CFA WRITE MULTIPLE WITHOUT ERASE
- CFA WRITE SECTORS WITHOUT ERASE
- DEVICE CONFIGURATION SET
- DOWNLOAD MICROCODE
- SECURITY DISABLE PASSWORD
- SECURITY ERASE UNIT
- SECURITY SET PASSWORD
- SECUITY UNLOCK
- SMART WRITE LOG
- WRITE BUFFER
- WRITE LOG EXT
- WRITE MULTIPLE
- WRITE MULTIPLE EXT
- WRITE MULTIPLE FUA EXT
- WRITE SECTOR(S)
- WRITE SECTOR(S) EXT
- WRITE STREAM PIO EXT

Execution of this class of command includes the transfer of one or more blocks of data from the host to the device. Figure 49 and the text following the figure describes the host states. Figure 50 and the text following the figure describes the device states.

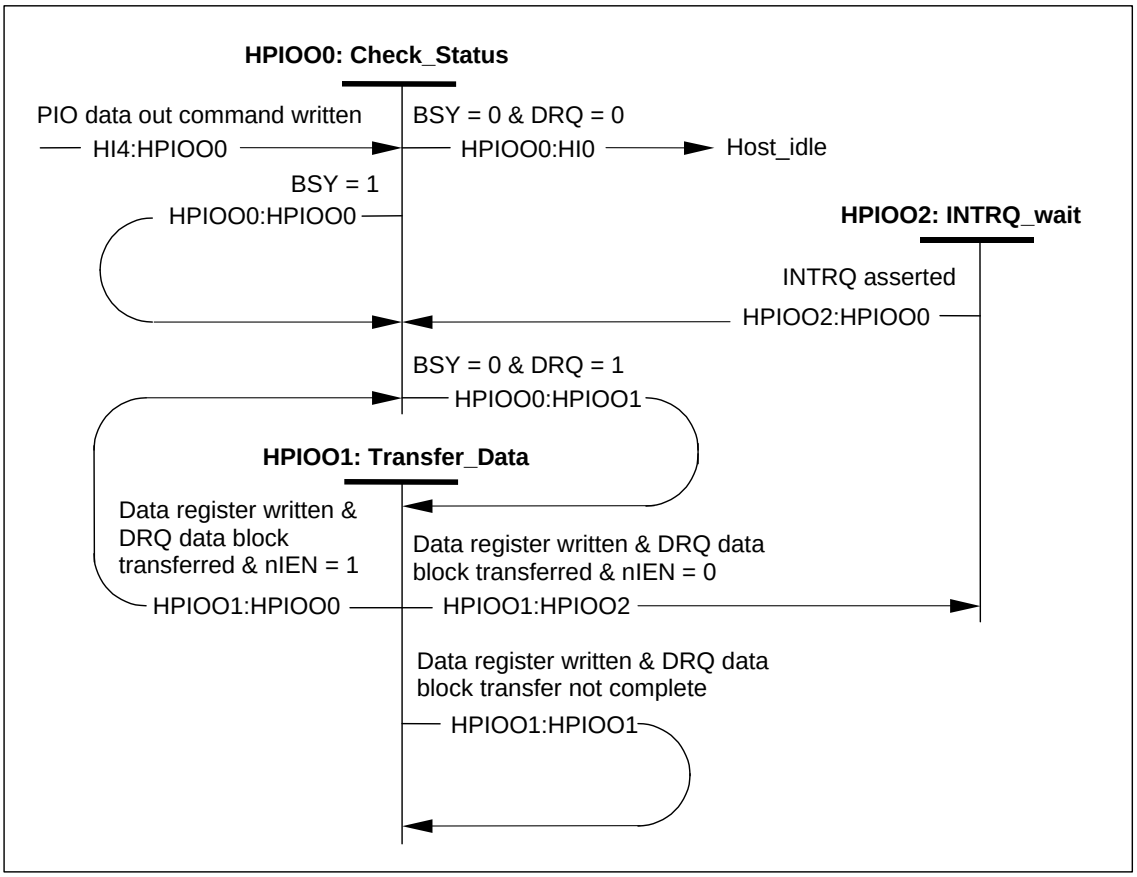



**Figure 49 – Host PIO data-Out state diagram**

**HPIOO0: Check_Status State:** This state is entered when the host has written a PIO data-out command to the device; when a DRQ data block has been written and nIEN is set to one; or when a DRQ data block has been written, nIEN is cleared zero, and INTRQ has been asserted.

When in this state, the host shall read the device Status register. When entering this state from the HI4 state, the host shall wait 400 ns before reading the Status register. When entering this state from the HPIOO1 state, the host shall wait one PIO transfer cycle time before reading the Status register. The wait may be accomplished by reading the Alternate Status register and ignoring the result.

**Transition HPIOO0:HI0:** When BSY is cleared to zero and DRQ is cleared to zero, then the device has completed the command and shall make a transition to the HI0: Host_Idle state (see Figure 41). If an error is reported, the host shall perform appropriate error recovery.

**Transition HPIOO0:HPIOO0:** When BSY is set to one and DRQ is cleared to zero, the host shall make a transition to the HPIOO0: Check_Status state.

**Transition HPIOO0:HPIOO1:** When BSY is cleared to zero and DRQ is set to one, the host shall make a transition to the HPIOO1: Transfer_Data state.

**HPIOO1: Transfer_Data State:** This state is entered when the BSY is cleared to zero, DRQ is set to one.

When in this state, the host shall write the device Data register to transfer data.

**Transition HPIOO1:HPIOO2:** When the host has written the device Data register, the DRQ data block has been transferred, and nIEN is cleared to zero, then the host shall make a transition to the HPIOO2: INTRQ_Wait state.

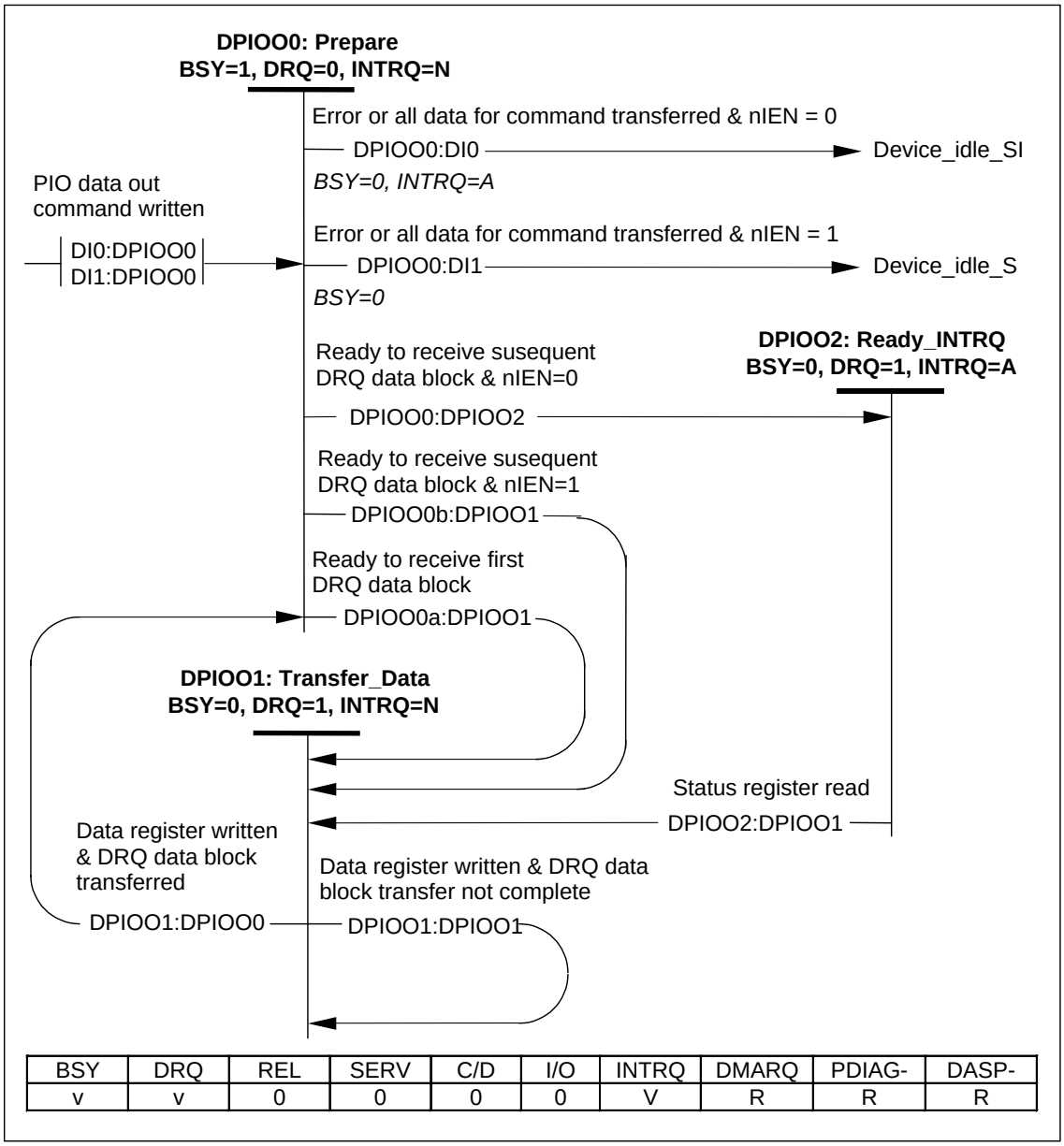
**Transition HPIOO1:HPIOO0:** When the host has written the device Data register, the DRQ data block has been transferred, and nIEN is set to one, then the host shall make a transition to the HPIOO0: Check_Status state.

**Transition HPIOO1:HPIOO1:** When the host has written the device Data register and the DRQ data block transfer has not completed, then the host shall make a transition to the HPIOO1: Transfer_Data state.

**HPIOO2: INTRQ_Wait State:** This state is entered when the host has completed a DRQ data block transfer and nIEN is cleared to zero.

When in this state, the host shall wait for INTRQ to be asserted.

**Transition HPIOO2:HPIOO0:** When INTRQ is asserted, the host shall make a transition to the HPIOO0: Check_Status state.




| BSY | DRQ | REL | SERV | C/D | I/O | INTRQ | DMARQ | PDIAG- | DASP- |
|---|---|---|---|---|---|---|---|---|---|
| v | v | 0 | 0 | 0 | 0 | V | R | R | R |

**Figure 50 – Device PIO data-Out state diagram**

**DPIOO0: Prepare State:** This state is entered when the device has a PIO data-out command written to the Command register or when a DRQ data block has been transferred.

When in this state, device shall set BSY to one within 400 ns of the writing of the Command register, shall clear DRQ to zero, and negate INTRQ. The device shall check for errors, determine if the data transfer is complete, and if not, prepare to receive the next DRQ data block.

**Transition DPIOO0a:DPIOO1:** When the device is ready to receive the first DRQ data block for a command, the device shall make a transition to the DPIOO1: Transfer_Data state.

**Transition DPIOO0b:DPIOO1:** When the device is ready to receive a subsequent DRQ data block for a command and nIEN is set to one, then the device shall set the interrupt pending and make a transition to the DPIOO1: Transfer_Data state.

**Transition DPIOO0:DPIOO2:** When the device is ready to receive a subsequent DRQ data block for a command and nIEN is cleared to zero, then the device shall set the interrupt pending and make a transition to the DPIOO2: Ready_INTRQ state.

**Transition DPIOO0:DI0:** When all data for the command has been transferred or an error occurs that causes the command to abort, and nIEN is cleared to zero, then the device shall set the interrupt pending, set appropriate error bits, clear BSY to zero, assert INTRQ, and make a transition to the DI0: Device_Idle_SI state (see Figure 43).

**Transition DPIOO0:DI1:** When all data for the command has been transferred or an error occurs that causes the command to abort, and nIEN is set to one, then the device shall set the interrupt pending, set appropriate error bits, clear BSY to zero, and make a transition to the DI1: Device_Idle_S state (see Figure 43).

**DPIOO1: Data_Transfer State:** This state is entered when the device is ready to receive a DRQ data block.

When in this state, BSY is cleared to zero, DRQ is set to one, INTRQ is negated, and the device recieves a data word in the Data register.

**Transition DPIOO1:DPIOO1:** When the Data register is written and transfer of the DRQ data block has not completed, then the device shall make a transition to the DPIOO1: Data_Transfer state.

**Transition DPIOO1:DPIOO0:** When the Data register is written and the transfer of the current DRQ data block has completed, then the device shall make a transition to the DPIOO0: Prepare state.

**DPIOO2: Ready_INTRQ State:** This state is entered when the device is ready to receive a DRQ data block and nIEN is cleared to zero.

When in this state, BSY is cleared to zero, DRQ is set to one, and INTRQ is asserted.

**Transition DPIOO2:DPIOO1:** When the Status register is read, the device shall clear the interrupt pending, negate INTRQ, and make a transition to the DPIOO1: Data_Transfer state.

## 11.7 DMA command protocol

This class includes:

- READ DMA
- READ DMA EXT
- READ STREAM DMA EXT
- WRITE DMA
- WRITE DMA EXT

- WRITE DMA FUA EXT
- WRITE STREAM DMA EXT

Execution of this class of command includes the transfer of one or more blocks of data from the host to the device or from the device to the host using DMA transfer. The host shall initialize the DMA channel prior to transferring data. A single interrupt is issued at the completion of the successful transfer of all data required by the command or when the transfer is aborted due to an error. Figure 51 and the text following the figure describes the host states. Figure 52 and the text following the figure describes the device states.

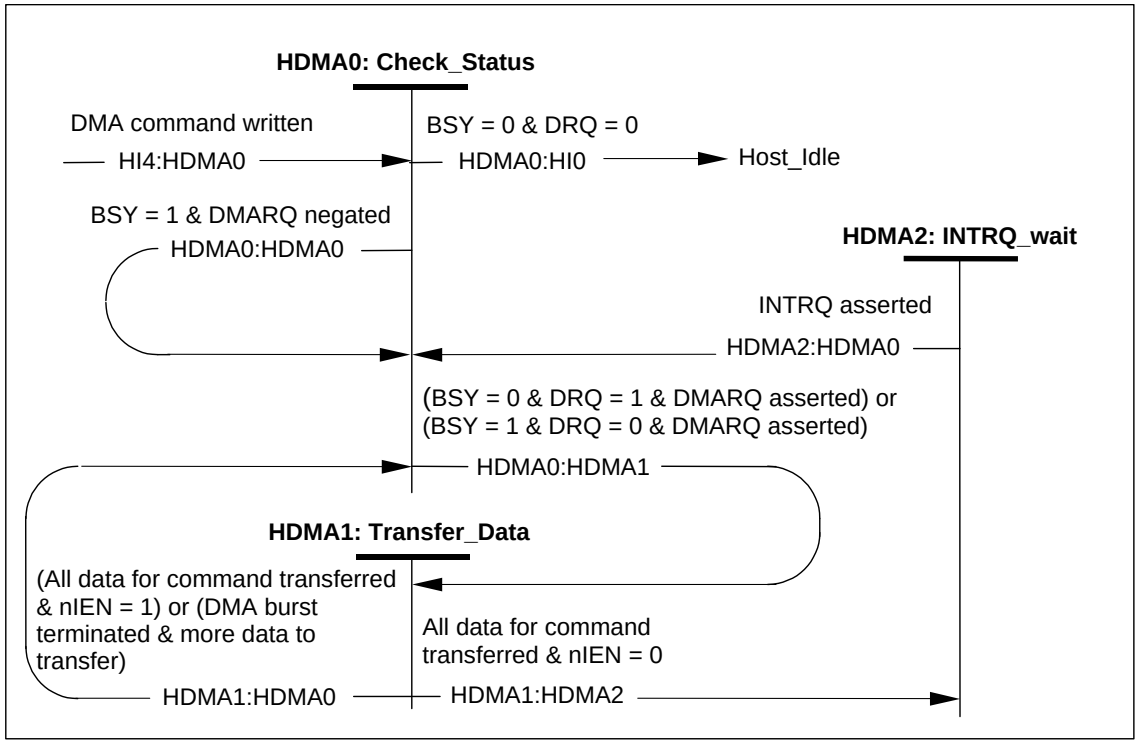



**Figure 51 – Host DMA state diagram**

**HDMA0: Check_Status State:** This state is entered when the host has written a DMA command to the device; when all data for the command has been transferred and nIEN is set to one; or when all data for the command has been transferred, nIEN is cleared zero, and INTRQ has been asserted.

When in this state, the host shall read the device Status register. When entering this state from the HI4 state, the host shall wait 400 ns before reading the Status register. When entering this state from the HDMA1 state, the host shall wait one PIO transfer cycle time before reading the Status register. The wait may be accomplished by reading the Alternate Status register and ignoring the result.

**Transition HDMA0:HI0:** When the BSY is cleared to zero and DRQ is cleared to zero, then the device has completed the command and shall make a transition to the HI0: Host_Idle state (see Figure 41). If an error is reported, the host shall perform appropriate error recovery.

**Transition HDMA0:HDMA0:** When BSY is set to one, DRQ is cleared to zero, and DMARQ is negated, then the host shall make a transition to the HDMA0: Check_Status state.

**Transition HDMA0:HDMA1:** When BSY is cleared to zero, DRQ is set to one, and DMARQ is asserted; or if BSY is set to one, DRQ is cleared to zero, and DMARQ is asserted, then the host shall make a transition to the HDMA1: Transfer_Data state. The host shall have set up the host DMA engine prior to making this transition.

**HDMA1: Transfer_Data State:** This state is entered when BSY is cleared to zero, DRQ is set to one, and DMARQ is asserted; or BSY is set to one, DRQ is cleared to zero, and DMARQ is asserted. The host shall have initialized the DMA channel prior to entering this state.

When in this state, the host shall perform the data transfer as described in the Multiword DMA timing or the Ultra DMA protocol.

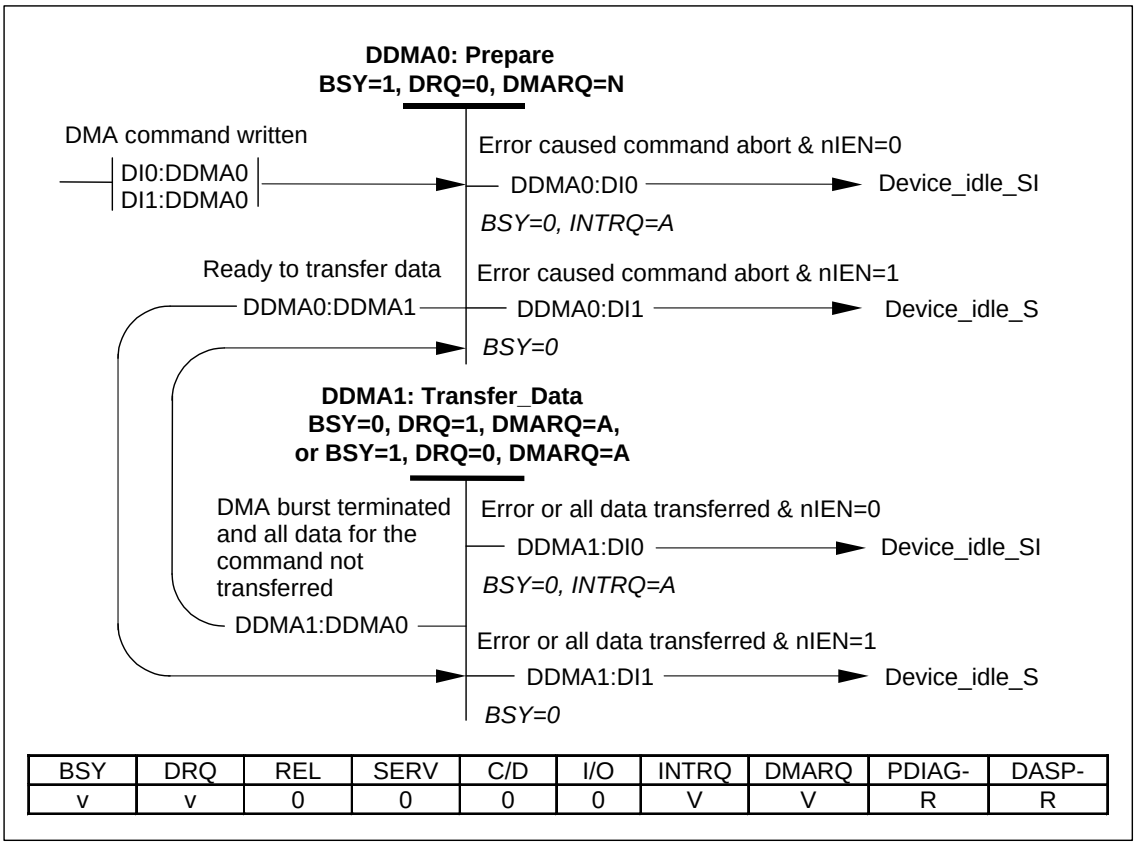
**Transition HDMA1:HDMA2:** When the host has transferred all data for the command and nIEN is cleared to zero, then the host shall make a transition to the HDMA2: INTRQ_Wait state.

**Transition HDMA1:HDMA0:** The host shall make a transition to the HDMA0: Check_Status state when 1) the host has transferred all data for the command and nIEN is set to one, or 2) the DMA burst has been terminated and all data for the command has not been transferred.

**HDMA2: INTRQ_Wait State:** This state is entered when the host has completed the transfer of all data for the command and nIEN is cleared to zero.

When in this state, the host shall wait for INTRQ to be asserted.

**Transition HDMA2:HDMA0:** When INTRQ is asserted, the host shall make a transition to the HDMA0: Check_Status state.




| BSY | DRQ | REL | SERV | C/D | I/O | INTRQ | DMARQ | PDIAG- | DASP- |
|---|---|---|---|---|---|---|---|---|---|
| v | v | 0 | 0 | 0 | 0 | V | V | R | R |

**Figure 52 – Device DMA state diagram**

**DDMA0: Prepare State:** This state is entered when the device has a DMA command written to the Command register.

When in this state, device shall set BSY to one, shall clear DRQ to zero, and negate INTRQ. The device shall check for errors, and prepare to transfer data.

**Transition DDMA0:DI0:** When an error is detected that causes the command to abort and nIEN is cleared to zero, the device shall set the appropriate error bits, enter the interrupt pending state, and make a transition to the DI0: Device_Idle_SI state (see Figure 43).

**Transition DDMA0:DI1:** When an error is detected that causes the command to abort and nIEN is set to one, then the device shall set the appropriate error bits, enter the interrupt pending state, and make a transition to the DI1: Device_Idle_S state (see Figure 43).

**Transition DDMA0:DDMA1:** When the device is ready transfer data for the command, the device shall make a transition to the DDMA1: Transfer_Data state.

**DDMA1: Data_Transfer State:** This state is entered when the device is ready to transfer data.

When in this state, BSY is cleared to zero, DRQ is set to one, and INTRQ is negated; or BSY is set to one, DRQ is cleared to zero, and INTRQ is negated. Data is transferred as decribed in Multiword DMA timing or Ultra DMA protocol.

**Transition DDMA1:DDMA0:** When the DMA burst is terminated and all data for the command has not been transferred, the device shall make a transition to the DDMA0: Prepare state.

**Transition DDMA1:DI0:** When the data transfer has completed or the device choses to abort the command due to an error and nIEN is cleared to zero, then the device shall set error bits if appropriate, enter the interrupt pending state, and make a transition to the DI0: Device_Idle_SI state (see Figure 43).

**Transition DDMA2:DI1:** When the data transfer has completed or the device choses to abort the command due to an error and nIEN is set to one, then the device shall set error bits if appropriate, enter the interrupt pending state, and make a transition to the DI1: Device_Idle_S state (see Figure 43).

## 11.8 PACKET command protocol

This class includes:

- PACKET

The PACKET command has a set of protocols for non-DMA data transfer commands and a set of protocols for DMA data transfer commands. Figure 53 and the text following the figure describes the host protocol for the PACKET command when non-data, PIO data-in, or PIO data-out is requested. Figure 54 and the text following the figure describes the device protocol for the PACKET command when non-data, PIO data-in, or PIO data-out is requested. Figure 55 and the text following the figure describes the host protocol for the PACKET command when DMA data transfer is requested. Figure 56 and the text following the figure describes the device protocol for the PACKET command when DMA data transfer is requested.

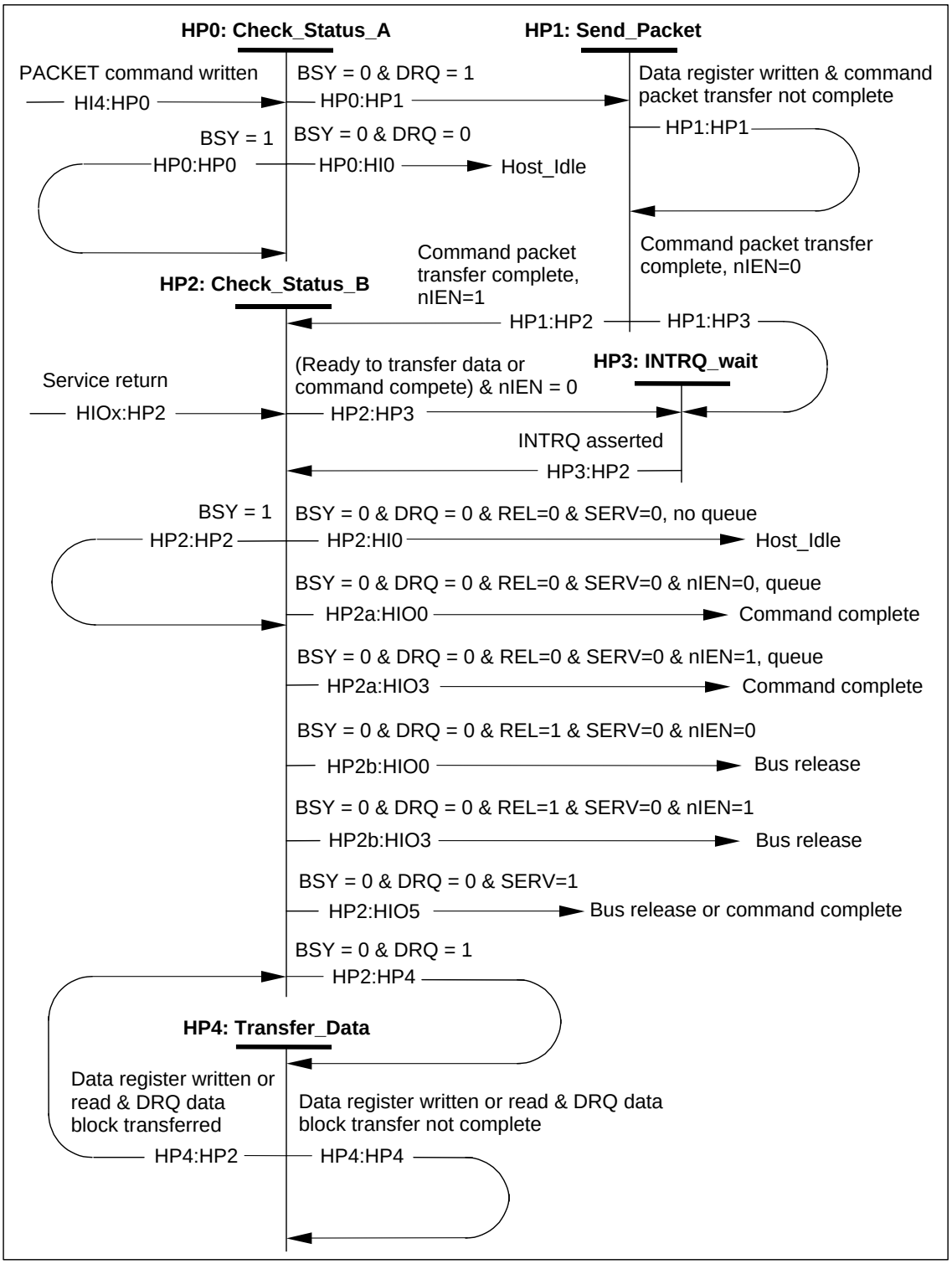



**Figure 53 – Host PACKET non-data and PIO data command state diagram**

**HP0: Check_Status_A State:** This state is entered when the host has written a PACKET command to the device.

When in this state, the host shall read the device Status register. When entering this state from the HI4 state, the host shall wait 400 ns before reading the Status register.

**Transition HP0:HP0:** When BSY is set to one, the host shall make a transition to the HP0: Check_Status_A state.

**Transition HP0:HP1:** When BSY is cleared to zero and DRQ is set to one, then the host shall make a transition to the HP1: Send_Packet state.

**Transition HP0:HI0:** When BSY is cleared to zero, DRQ is cleared to zero, REL is cleared to zero, and SERV is cleared to zero, then the command is completed and the host shall make a transition to the HI0: Host_Idle state (see Figure 41). If an error is reported, the host shall perform appropriate error recovery.

**HP1: Send_Packet State:** This state is entered when BSY is cleared to zero, DRQ is set to one.

When in this state, the host shall write a byte of the command packet to the Data register.

**Transition HP1:HP1:** When the Data register has been written and the writing of the command packet is not completed, the host shall make a transition to the HP1: Send_Packet state.

**Transition HP1:HP2:** When the Data register has been written, the writing of the command packet is completed, and nIEN is set to one, the host shall make a transition to the HP2: Check_Status_B state.

**Transition HP1:HP3:** When the Data register has been written, the writing of the command packet is completed, and nIEN is cleared to zero, the host shall make a transition to the HP3: INTRQ wait state.

**HP2: Check_Status_B State:** This state is entered when the host has written the command packet to the device, when INTRQ has been asserted, when a DRQ data block has been transferred, or from a service return.

When in this state, the host shall read the device Status register. When entering this state from the HP1 or HP4 state, the host shall wait one PIO transfer cycle time before reading the Status register. The wait may be accomplished by reading the Alternate Status register and ignoring the result.

**Transition HP2:HP2:** When BSY is set to one, and DRQ is cleared to zero, the host shall make a transition to the HP2: Check_Status_B state.

**Transition HP2:HP3:** When the host is ready to transfer data or the command is complete, and nIEN is cleared to zero, then the host shall make a transition to the HP3: INTRQ_Wait state.

**Transition HP2:HP4:** When BSY is cleared to zero and DRQ is set to one, then the host shall make a transition to the HP4: Transfer_Data state.

**Transition HP2:HI0:** When BSY is cleared to zero, DRQ is cleared to zero, REL is cleared to zero, SERV is cleared to zero, and the device queue is empty, then the command is completed and the host shall make a transition to the HI0: Host_Idle state (see Figure 41). If an error is reported, the host shall perform appropriate error recovery.

**Transition HP2a:HIO0:** When BSY is cleared to zero, DRQ is cleared to zero, REL is cleared to zero, SERV is cleared to zero, nIEN is cleared to zero, and the device has a queue of released commands, then the command is completed and the host shall make a transition to the HIO0: Command completed state (see Figure 42). If an error is reported, the host shall perform appropriate error recovery.

**Transition HP2a:HIO3:** When BSY is cleared to zero, DRQ is cleared to zero, REL is cleared to zero, SERV is cleared to zero, nIEN is set to one, and the device has a queue of released commands, then the command is completed and the host shall make a transition to the HIO3: Command completed state (see Figure 42). If an error is reported, the host shall perform appropriate error recovery.

**Transitions HP2b:HIO0:** When BSY is cleared to zero, DRQ is cleared to zero, REL is set to one, SERV is cleared to zero, and nIEN is cleared to zero, then the host shall make a transition to the HIO0: INTRQ_wait_A state (see Figure 42). The bus has been released.

**Transitions HP2b:HIO3:** When BSY is cleared to zero, DRQ is cleared to zero, REL is set to one, SERV is cleared to zero, and nIEN is set to one, then the host shall make a transition to the HIO3: Check_status_A state (see Figure 42). The bus has been released.

**Transitions HP2:HIO5:** When BSY is cleared to zero, DRQ is cleared to zero, and SERV is set to one, then the host shall make a transition to the HIO5: Write_SERVICE state (see Figure 42). The command is completed or the bus has been released, and another queued command is ready for service. If an error is reported, the host shall perform appropriate error recovery.

**HP3: INTRQ_Wait State:** This state is entered when the command packet has been transmitted, the host is ready to transfer data or when the command has completed, and nIEN is cleared to zero.

When in this state, the host shall wait for INTRQ to be asserted.

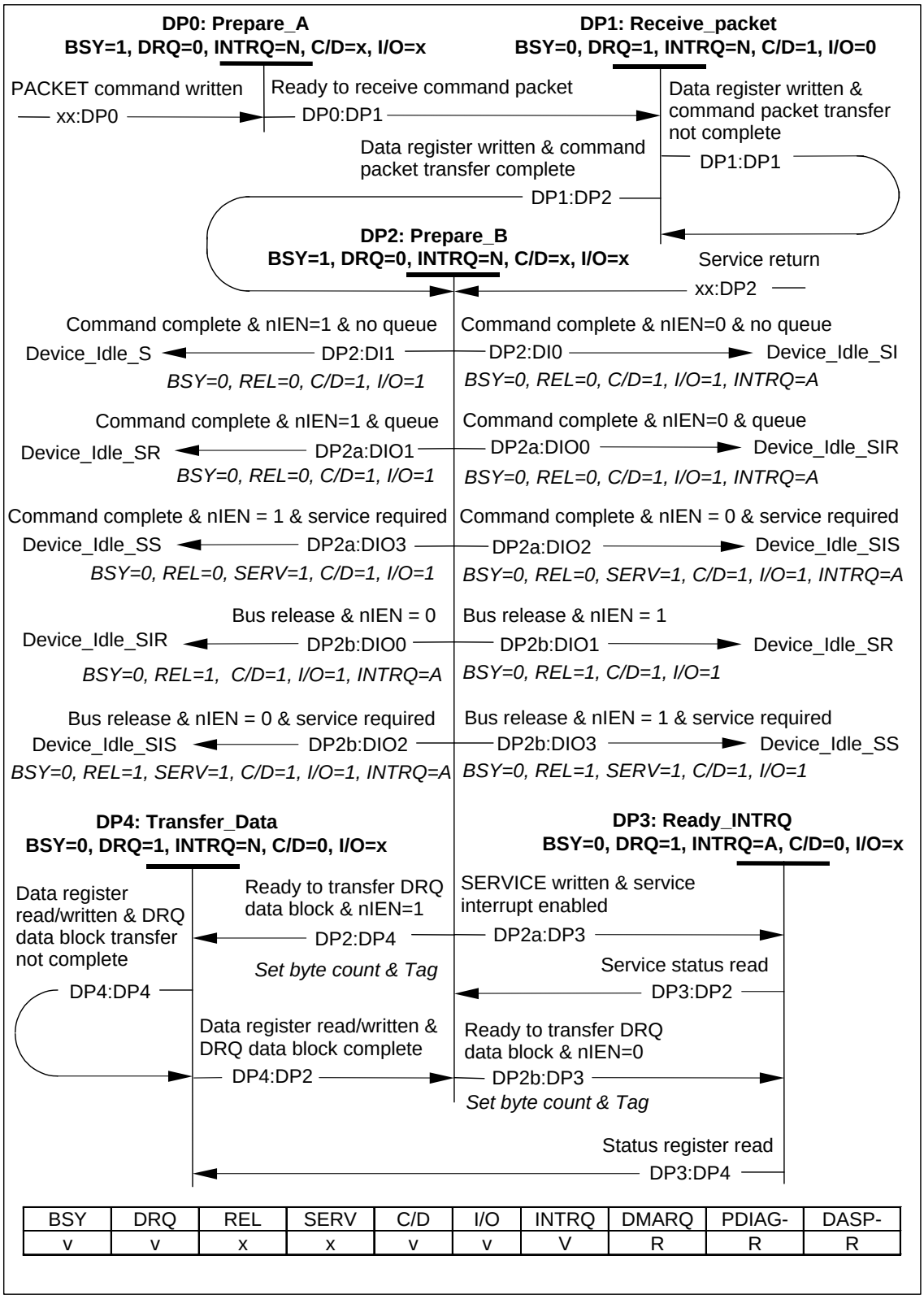
**Transition HP3:HP2:** When INTRQ is asserted, the host shall make a transition to the HP2: Check_Status_B state.

**HP4: Transfer_Data State:** This state is entered when BSY is cleared to zero, DRQ is set to one, and C/D is cleared to zero.

When in this state, the host shall read the byte count then read or write the device Data register to transfer data. If the bus has been released, the host shall read the Sector Count register to determine the Tag for the queued command to be executed.

**Transition HP4:HP2:** When the host has read or written the device Data register and the DRQ data block has been transferred, then the host shall make a transition to the HP2: Check_Status_B state.

**Transition HP4:HP4:** When the host has read or written the device status register and the DRQ data block transfer has not completed, then the host shall make a transition to the HP4: Transfer_Data state.




| BSY | DRQ | REL | SERV | C/D | I/O | INTRQ | DMARQ | PDIAG- | DASP- |
|---|---|---|---|---|---|---|---|---|---|
| v | v | x | x | v | v | V | R | R | R |

**Figure 54 – Device PACKET non-data and PIO data command state diagram**

.

**DP0: Prepare_A State:** This state is entered when the device has a PACKET written to the Command register.

When in this state, device shall set BSY to one, clear DRQ to zero, and negate INTRQ within 400 ns of the receipt of the command and shall prepare to receive a command packet. If the command is a queued command, the device shall verify that the Tag is valid.

**Transition DP0:DP1:** When the device is ready to receive the command packet for a command, the device shall make a transition to the DP1: Receive_Packet state.

**DP1: Receive_Packet State:** This state is entered when the device is ready to receive the command packet.

When in this state, BSY is cleared to zero, DRQ is set to one, INTRQ is negated, C/D is set to one, I/O is cleared to zero, and REL is cleared to zero. When in this state, the device Data register is written.

**Transition DP1:DP1:** If the Data register is written and the entire command packet has not been received, then the device shall make a transition to the DP1: Receive_Packet state.

**Transition DP1:DP2:** When the Data register is written and the entire command packet has been received, then the device shall make a transition to the DP2: Prepare_B state.

**DP2: Prepare_B State:** This state is entered when the command packet has been received or from a Service return.

When in this state, device shall set BSY to one, clear DRQ to zero, and negate INTRQ. Non-data transfer commands shall be executed while in this state. For data transfer commands, the device shall check for errors, determine if the data transfer is complete, and if not, prepare to transfer the next DRQ data block.

If the command is overlapped and the release interrupt is enabled, the device shall bus release as soon as the command packet has been received.

**Transition DP2:DP4:** When the device is ready to transfer a DRQ data block for a command and nIEN is set to one, then the device shall set the command Tag and byte count, set the interrupt pending, and make a transition to the DP4: Transfer_Data state.

**Transition DP2b:DP3:** When the device is ready to transfer a DRQ data block for a command and nIEN is cleared to zero, then the device shall set the command Tag and byte count, set the interrupt pending, and make a transition to the DP3: Ready_INTRQ state.

**Transition DP2a:DP3:** When the service interrupt is enabled and the device has SERVICE written to the Command register, then the device shall set the command Tag and byte count and make a transition to the DP3: Ready_INTRQ state.

**Transition DP2:DI0:** When the command has completed or an error occurs that causes the command to abort, the device has no other command released, and nIEN is cleared to zero, then the device shall set the interrupt pending, set appropriate error bits, set C/D and I/O to one, clear BSY to zero, and make a transition to the DI0: Device_Idle_SI state (see Figure 43).

**Transition DP2:DI1:** When the command has completed or an error occurs that causes the command to abort, the device has no other command released, and nIEN is set to one, then the device shall set appropriate error bits, set C/D and I/O to one, clear BSY to zero, and make a transition to the DI1: Device_Idle_S state (see Figure 43).

**Transition DP2a:DIO0:** When the command has completed or an error occurs that causes the command to abort, the device has another command released but not ready for service, and nIEN is cleared to zero, then the device shall set the interrupt pending, set appropriate error bits, set C/D and I/O to one, clear BSY to zero, and make a transition to the DIO0: Device_Idle_SIR state (see Figure 44).

**Transition DP2a:DIO1:** When the command has completed or an error occurs that causes the command to abort, the device has another command released but not ready for service, and nIEN is set to one, then the device shall set appropriate error bits, set C/D and I/O to one, clear BSY to zero, and make a transition to the DIO1: Device_Idle_SR state (see Figure 44).

**Transition DP2a:DIO2:** When the command has completed or an error occurs that causes the command to abort, the device has another command ready for service, and nIEN is cleared to zero, then the device shall set the interrupt pending, set appropriate error bits, set C/D and I/O to one, set SERV to one, clear BSY to zero, and make a transition to the DIO2: Device_Idle_SIS state (see Figure 44).

**Transition DP2a:DIO3:** When the command has completed or an error occurs that causes the command to abort, the device has another command ready for service, and nIEN is set to one, then the device shall set appropriate error bits, set C/D and I/O to one, set SERV to one, clear BSY to zero, and make a transition to the DIO3: Device_Idle_SS state (see Figure 44).

**Transition DP2b:DIO0:** When the command is released and nIEN is cleared to zero, then the device shall set the interrupt pending, set appropriate error bits, set C/D and I/O to one, set REL to one, clear BSY to zero, and make a transition to the DIO0: Device_Idle_SIR state (see Figure 44).

**Transition DP2b:DIO1:** When the command is released and nIEN is set to one, then the device shall set appropriate error bits, set C/D and I/O to one, set REL to one, clear BSY to zero, and make a transition to the DIO1: Device_Idle_SR state (see Figure 44).

**Transition DP2b:DIO2:** When the command is released, the device has another command ready for service, and nIEN is cleared to zero, then the device shall set the interrupt pending, set appropriate error bits, set C/D and I/O to one, set REL to one, set SERV to one, clear BSY to zero, and make a transition to the DIO2: Device_Idle_SIS state (see Figure 44).

**Transition DP2b:DIO3:** When the command is released, the device has another command ready for service, and nIEN is set to one, then the device shall set appropriate error bits, set C/D and I/O to one, set REL to one, set SERV to one, clear BSY to zero, and make a transition to the DIO3: Device_Idle_SS state (see Figure 44).

**DP3: Ready_INTRQ State:** This state is entered when the device is ready to transfer a DRQ data block and nIEN is cleared to zero. This state is entered to interrupt upon receipt of a SERVICE command when service interrupt is enabled.

When in this state, BSY is cleared to zero, DRQ is set to one, INTRQ is asserted, C/D is cleared to zero, and I/O is set to one for PIO data-out or cleared to zero for PIO data-in.

**Transition DP3:DP2:** When the Status register is read to respond to a service interrupt, the device shall make a transition to the DP2: Prepare_B state.

**Transition DP3:DP4:** When the Status register is read when the device is ready to transfer data, then the device shall clear the interrupt pending, negate INTRQ, and make a transition to the DP4: Data_Transfer state.

**DP4: Data_Transfer State:** This state is entered when the device is ready to transfer a DRQ data block.

When in this state, BSY is cleared to zero, DRQ is set to one, INTRQ is negated, C/D is cleared to zero, I/O is set to one for PIO data-out or cleared to zero for PIO data-in, and a data word is read/written in the Data register.

**Transition DP4:DP4:** When the Data register is read/written and transfer of the DRQ data block has not completed, then the device shall make a transition to the DP4: Data_Transfer state.

**Transition DP4:DP2:** When the Data register is read/written and the transfer of the current DRQ data block has completed, then the device shall make a transition to the DP2: Prepare_B state.

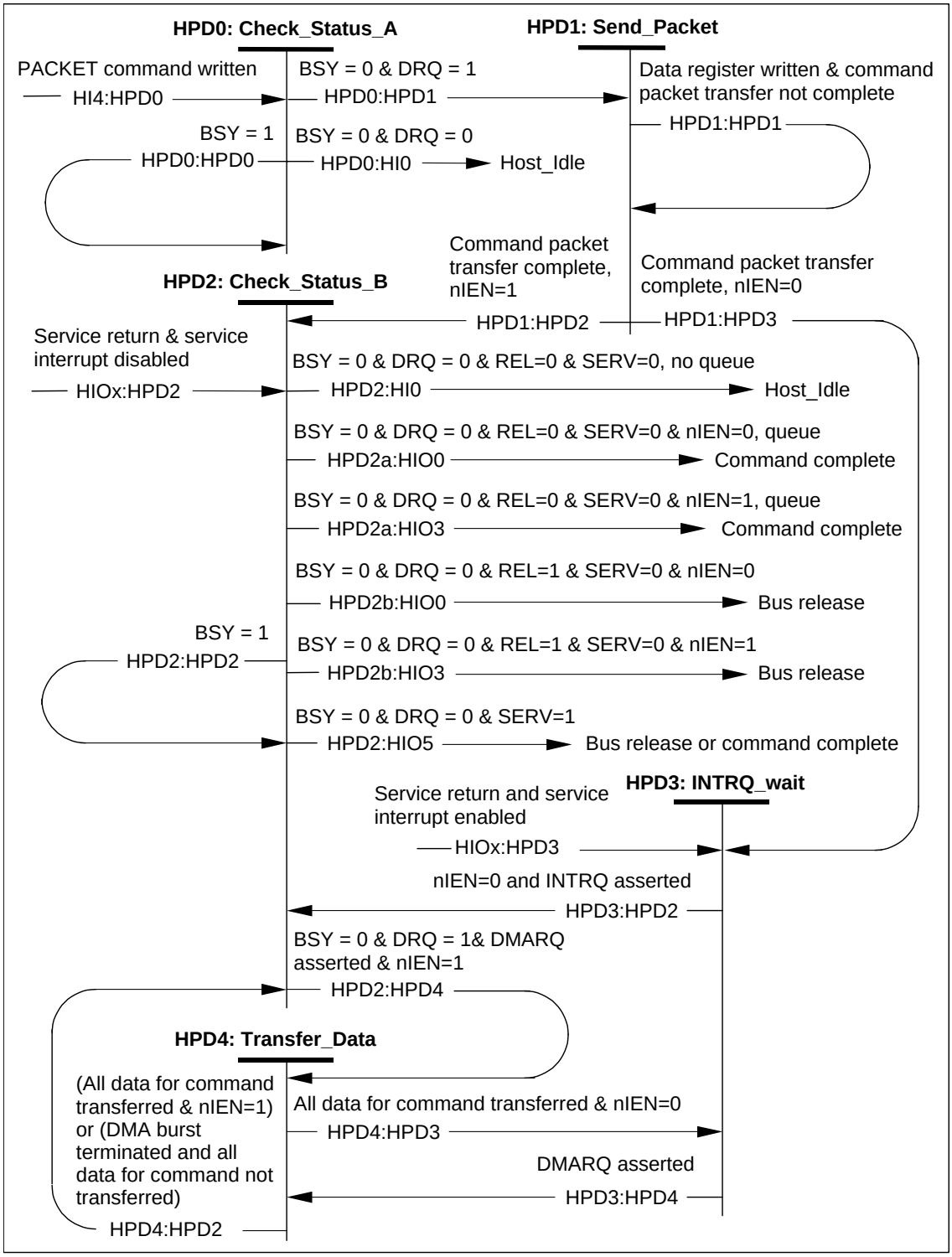



**Figure 55 – Host PACKET DMA command state diagram**

**HPD0: Check_Status_A State:** This state is entered when the host has written a PACKET command to the device.

When in this state, the host shall read the device Status register. When entering this state from the HI4 state, the host shall wait 400 ns before reading the Status register.

**Transition HPD0:HPD0:** When BSY is set to one, the host shall make a transition to the HPD0: Check_Status_A state.

**Transition HPD0:HPD1:** When BSY is cleared to zero and DRQ is set to one, then the host shall make a transition to the HPD1: Send_Packet state.

**Transition HPD0:HI0:** When BSY is cleared to zero, DRQ is cleared to zero, REL is cleared to zero, and SERV is cleared to zero, then the command is completed and the host shall make a transition to the HI0: Host_Idle state (see Figure 41). If an error is reported, the host shall perform appropriate error recovery.

**HPD1: Send_Packet State:** This state is entered when BSY is cleared to zero, DRQ is set to one.

When in this state, the host shall write a byte of the command packet to the Data register.

**Transition HPD1:HPD1:** When the Data register has been written and the writing of the command packet is not completed, the host shall make a transition to the HPD1: Send_Packet state.

**Transition HPD1:HPD2:** When the Data register has been written, the writing of the command packet is completed, and nIEN is set to one, the host shall make a transition to the HPD2: Check_Status_B state.

**Transition HPD1:HPD3:** When the Data register has been written, the writing of the command packet is completed, and nIEN is cleared to zero, the host shall make a transition to the HPD3: INTRQ wait state.

**HPD2: Check_Status_B State:** This state is entered when the host has written the command packet to the device, when INTRQ has been asserted, when a DRQ data block has been transferred, or from a service return when the service interrupt is disabled.

When in this state, the host shall read the device Status register. When entering this state from the HPD1 or HPD4 state, the host shall wait one PIO transfer cycle time before reading the Status register. The wait may be accomplished by reading the Alternate Status register and ignoring the result.

**Transition HPD2:HPD2:** When BSY is set to one, and DRQ is cleared to zero, the host shall make a transition to the HPD2: Check_Status_B state.

**Transition HPD2:HPD4:** When BSY is cleared to zero, DRQ is set to one, and DMARQ is asserted and nIEN=1, then the host shall make a transition to the HPD4: Transfer_Data state. The host shall have set up the DMA engine before this transition.

**Transition HPD2:HI0:** When BSY is cleared to zero, DRQ is cleared to zero, REL is cleared to zero, SERV is cleared to zero, and the device queue is empty, then the command is completed and the host shall make a transition to the HI0: Host_Idle state (see Figure 41). If an error is reported, the host shall perform appropriate error recovery.

**Transition HPD2a:HIO0:** When BSY is cleared to zero, DRQ is cleared to zero, REL is cleared to zero, SERV is cleared to zero, nIEN is cleared to zero, and the device has a queue of released commands, then the command is completed and the host shall make a transition to the HIO0: Command completed state (see Figure 42). If an error is reported, the host shall perform appropriate error recovery.

**Transition HPD2a:HIO3:** When BSY is cleared to zero, DRQ is cleared to zero, REL is cleared to zero, SERV is cleared to zero, nIEN is set to one, and the device has a queue of released commands, then the command is completed and the host shall make a transition to the HIO3: Command completed state (see Figure 42). If an error is reported, the host shall perform appropriate error recovery.

**Transition HPD2b:HIO0:** When BSY is cleared to zero, DRQ is cleared to zero, REL is set to one, SERV is cleared to zero, and nIEN is cleared to zero, then the host shall make a transition to the HIO0: INTRQ_wait_A state (see Figure 42). The bus has been released.

**Transition HPD2b:HIO3:** When BSY is cleared to zero, DRQ is cleared to zero, REL is set to one, SERV is cleared to zero, and nIEN is set to one, then the host shall make a transition to the HIO3: Check_status_A state (see Figure 42). The bus has been released.

**Transition HPD2:HIO5:** When BSY is cleared to zero, DRQ is cleared to zero, and SERV is set to one, then the host shall make a transition to the HIO5: Write_SERVICE state (see Figure 42). The command is completed or the bus has been released, and another queued command is ready for service. If an error is reported, the host shall perform appropriate error recovery.

**HPD3: INTRQ_Wait State:** This state is entered when the command packet has been transmitted, when a service return is issued and the service interrupt is enabled, or when the command has completed and nIEN is cleared to zero.

When in this state, the host shall wait for INTRQ to be asserted if nIEN=0 or DMARQ if nIEN=1.

**Transition HPD3:HPD2:** When INTRQ is asserted and nIEN=0, the host shall make a transition to the HPD2: Check_Status_B state.

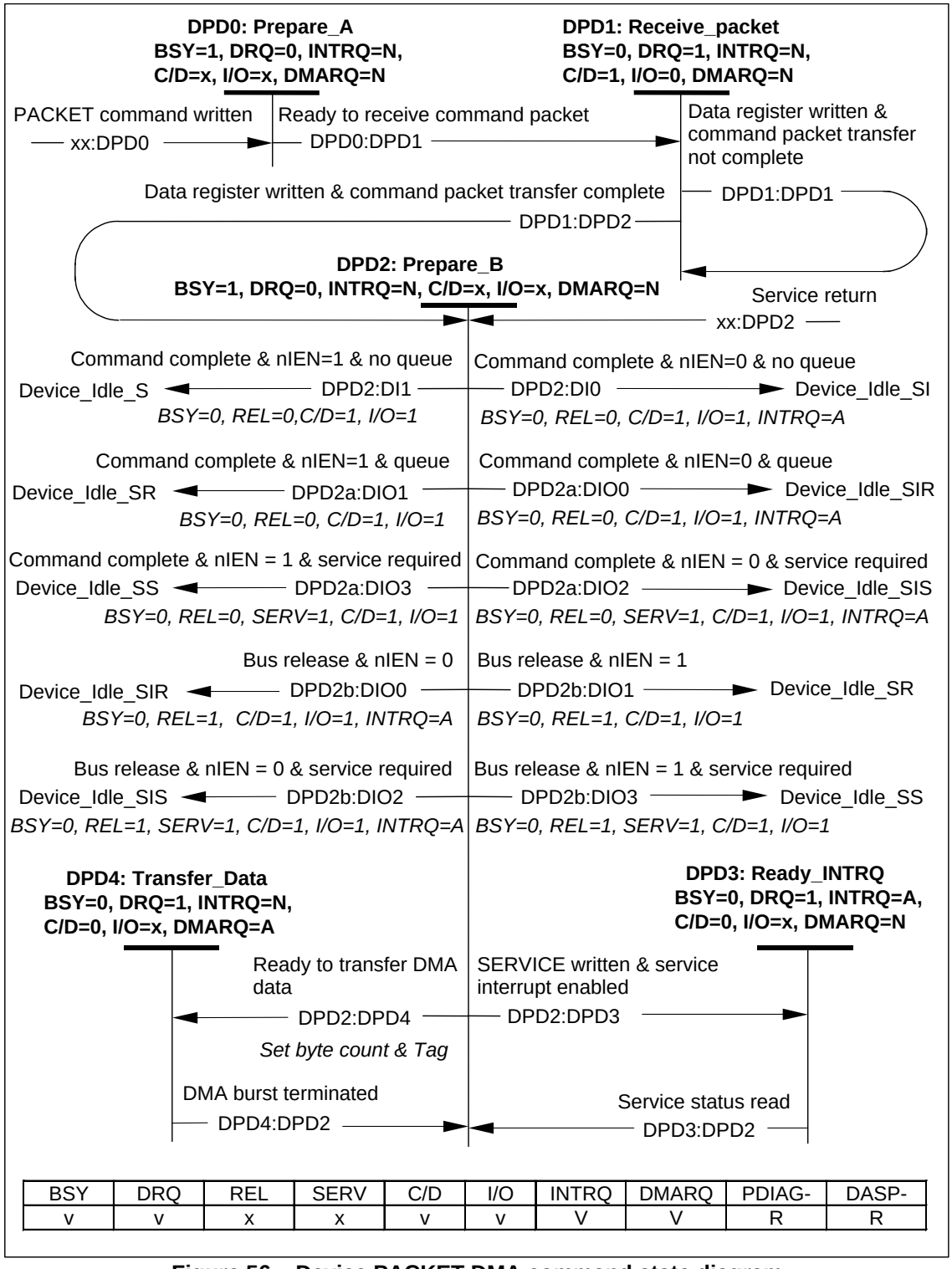
**Transition HPD3:HPD4:** When DMARQ is asserted, the host shall make a transition to the HPD4: Transfer_Data state.

**HPD4: Transfer_Data State:** This state is entered when BSY is cleared to zero, DRQ is set to one, and DMARQ is asserted.

When in this state, the host shall read or write the device Data port to transfer data. If the bus has been released, the host shall read the Sector Count register to determine the Tag for the queued command to be executed.

**Transition HPD4:HPD2:** The host shall make a transition to the HPD2: Check_Status_B state when 1) the host has transferred all data for the command and nIEN is set to one, or 2) the DMA burst has been terminated and all data for the command has not been transferred.

**Transition HPD4:HPD3:** When all data for the request has been transferred and nIEN is cleared to zero, then the host shall make a transition to the HPD3: INTRQ_wait state.




| BSY | DRQ | REL | SERV | C/D | I/O | INTRQ | DMARQ | PDIAG- | DASP- |
|---|---|---|---|---|---|---|---|---|---|
| v | v | x | x | v | v | V | V | R | R |

**Figure 56 – Device PACKET DMA command state diagram**

**DPD0: Prepare_A State:** This state is entered when the device has a PACKET written to the Command register.

When in this state, device shall set BSY to one, clear DRQ to zero, and negate INTRQ within 400 ns of the receipt of the command and shall prepare to receive a command packet. If the command is a queued command, the device shall verify that the Tag is valid.

**Transition DPD0:DPD1:** When the device is ready to receive the command packet for a command, the device shall make a transition to the DPD1: Receive_Packet state.

**DPD1: Receive_Packet State:** This state is entered when the device is ready to receive the command packet.

When in this state, BSY is cleared to zero, DRQ is set to one, INTRQ is negated, C/D is set to one, I/O is cleared to zero, and REL is cleared to zero. When in this state, the device Data register is written.

**Transition DPD1:DPD1:** If the Data register is written and the entire command packet has not been received, then the device shall make a transition to the DPD1: Receive_Packet state.

**Transition DPD1:DPD2:** When the Data register is written and the entire command packet has been received, then the device shall make a transition to the DPD2: Prepare_B state.

**DPD2: Prepare_B State:** This state is entered when the command packet has been received or from a Service return.

When in this state, device shall set BSY to one, clear DRQ to zero, and negate INTRQ. The device shall check for errors, determine if the data transfer is complete, and if not, prepare to transfer the DMA data.

If the command is overlapped and the release interrupt is enabled, the device shall bus release as soon as the command packet has been received.

**Transition DPD2:DPD4:** When the device is ready to transfer DMA data for a command and nIEN is set to one, then the device shall set the command Tag and byte count, set the interrupt pending, and make a transition to the DPD4: Transfer_Data state.

**Transition DPD2:DPD3:** When the service interrupt is enabled and the device has SERVICE written to the Command register, then the device shall set the command Tag and byte count and make a transition to the DPD3: Ready_INTRQ state.

**Transition DPD2:DI0:** When the command has completed or an error occurs that causes the command to abort, the device has no other command released, and nIEN is cleared to zero, then the device shall set the interrupt pending, set appropriate error bits, set C/D and I/O to one, clear BSY to zero, and make a transition to the DI0: Device_Idle_SI state (see Figure 43).

**Transition DPD2:DI1:** When the command has completed or an error occurs that causes the command to abort, the device has no other command released, and nIEN is set to one, then the device shall set appropriate error bits, set C/D and I/O to one, clear BSY to zero, and make a transition to the DI1: Device_Idle_S state (see Figure 43).

**Transition DPD2a:DIO0:** When the command has completed or an error occurs that causes the command to abort, the device has another command released but not ready for service, and nIEN is cleared to zero, then the device shall set the interrupt pending, set appropriate error bits, set C/D and I/O to one, clear BSY to zero, and make a transition to the DIO0: Device_Idle_SIR state (see Figure 44).

**Transition DPD2a:DIO1:** When the command has completed or an error occurs that causes the command to abort, the device has another command released but not ready for service, and nIEN is set to one, then the device shall, set appropriate error bits, set C/D and I/O to one, clear BSY to zero, and make a transition to the DIO1: Device_Idle_SR state (see Figure 44).

**Transition DPD2a:DIO2:** When the command has completed or an error occurs that causes the command to abort, the device has another command ready for service, and nIEN is cleared to zero, then the device shall set the interrupt pending, set appropriate error bits, set C/D and I/O to one, set SERV to one, clear BSY to zero, and make a transition to the DIO2: Device_Idle_SIS state (see Figure 44).

**Transition DPD2a:DIO3:** When the command has completed or an error occurs that causes the command to abort, the device has another command ready for service, and nIEN is set to one, then the device shall set appropriate error bits, set C/D and I/O to one, set SERV to one, clear BSY to zero, and make a transition to the DIO3: Device_Idle_SS state (see Figure 44).

**Transition DPD2b:DIO0:** When the command is released and nIEN is cleared to zero, then the device shall set the interrupt pending, set appropriate error bits, set C/D and I/O to one, set REL to one, clear BSY to zero, and make a transition to the DIO0: Device_Idle_SIR state (see Figure 44).

**Transition DPD2b:DIO1:** When the command is released and nIEN is set to one, then the device shall, set appropriate error bits, set C/D and I/O to one, set REL to one, clear BSY to zero, and make a transition to the DIO1: Device_Idle_SR state (see Figure 44).

**Transition DPD2b:DIO2:** When the is released, the device has another command ready for service, and nIEN is cleared to zero, then the device shall set the interrupt pending, set appropriate error bits, set C/D and I/O to one, set REL to one, set SERV to one, clear BSY to zero, and make a transition to the DIO2: Device_Idle_SIS state (see Figure 44).

**Transition DPD2b:DIO3:** When the command is released, the device has another command ready for service, and nIEN is set to one, then the device shall set appropriate error bits, set C/D and I/O to one, set REL to one, set SERV to one, clear BSY to zero, and make a transition to the DIO3: Device_Idle_SS state (see Figure 44).

**DPD3: Ready_INTRQ State:** This state is entered upon receipt of a SERVICE command when service interrupt is enabled.

When in this state, BSY is cleared to zero, DRQ is set to one, INTRQ is asserted, C/D is cleared to zero, and I/O is set to one for PIO data-out or cleared to zero for PIO data-in.

**Transition DPD3:DPD2:** When the Status register is read to respond to a service interrupt, the device shall make a transition to the DPD2: Prepare_B state.

**DPD4: Data_Transfer State:** This state is entered when the device is ready to transfer DMA data.

When in this state, BSY is cleared to zero, DRQ is set to one, INTRQ is negated, C/D is cleared to zero, I/O is set to one for data-out or cleared to zero for data-in, DMARQ is asserted, and data is transferred as described in Multiword DMA timing or Ultra DMA protocol.

**Transition DPD4:DPD2:** When the DMA burst is terminated, the device shall make a transition to the DPD2: Prepare_B state. All of the data for the command may not have been transferred.

## 11.9 READ/WRITE DMA QUEUED command protocol

This class includes:

- READ DMA QUEUED
- READ DMA QUEUED EXT
- WRITE DMA QUEUED
- WRITE DMA QUEUED EXT
- WRITE DMA QUEUED FUA EXT

Execution of this class of command includes the transfer of one or more blocks of data from the host to the device or from the device to the host using DMA transfer. All data for the command may be transferred without a bus release between the command receipt and the data transfer. This command may bus release before transferring data. The host shall initialize the DMA channel prior to transferring data. When data transfer is begun, all data for the request shall be transferred without a bus release. Figure 57 and the text

following the figure describes the host states. Figure 58 and the text following the figure describes the device states.

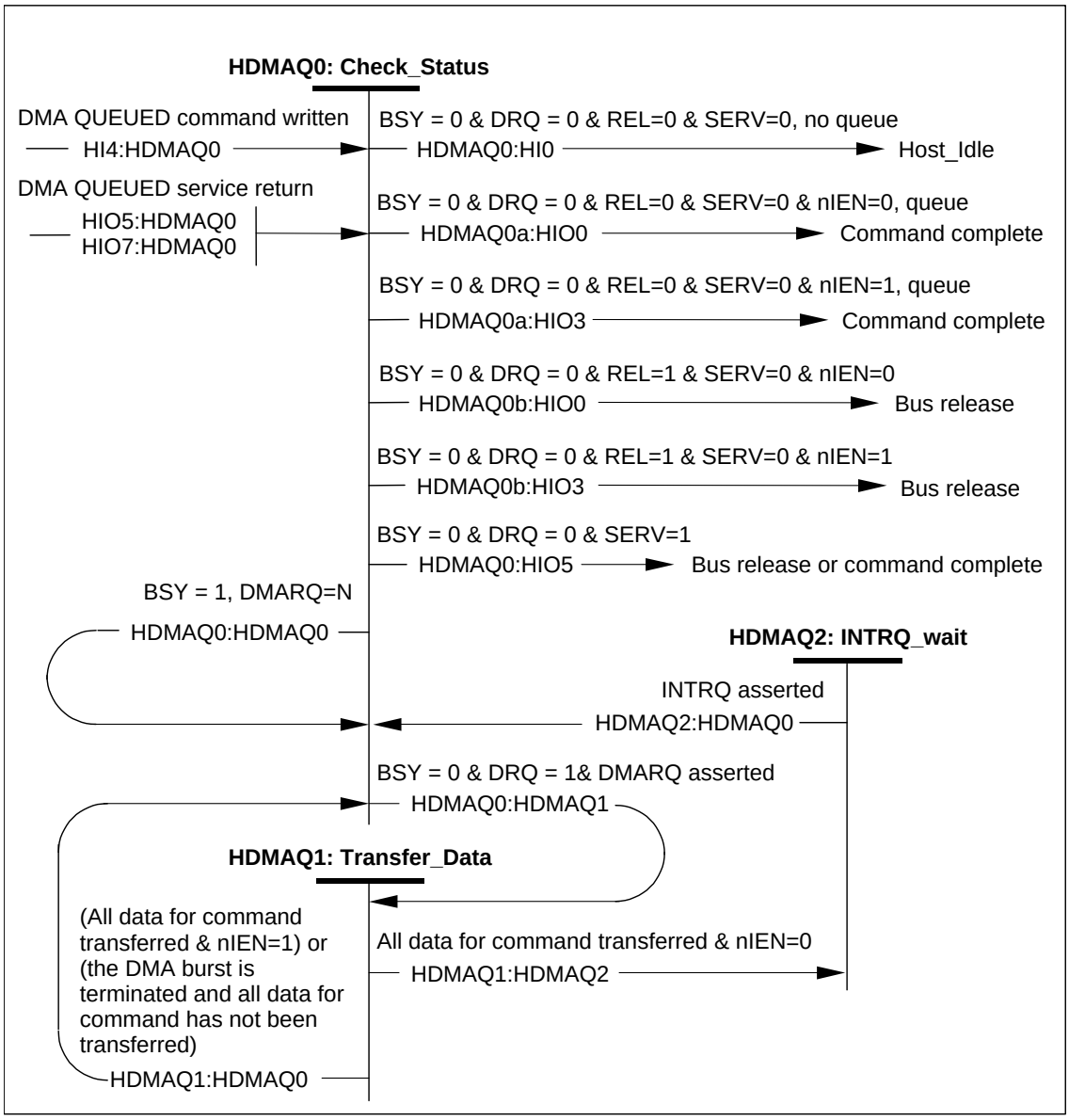



**Figure 57 – Host DMA QUEUED state diagram**

**HDMAQ0: Check_Status State:** This state is entered when the host has written a READ/WRITE DMA QUEUED command to the device, when all data for the command has been transferred and nIEN is set to one, or when all data for the command has been transferred, nIEN is cleared to zero, and INTRQ has been asserted. It is also entered when the SERVICE command has been written to continue execution of a bus released command.

When in this state, the host shall read the device Status register. When entering this state from the HI4, HIO5, or HIO7 state, the host shall wait 400 ns before reading the Status register. When entering this state from the HDMAQ1 state, the host shall wait one PIO transfer cycle time before reading the Status register. The wait may be accomplished by reading the Alternate Status register and ignoring the result. When entering this state from the DMA QUEUED service return, the host shall check the Tag for the command to be serviced before making a transition to transfer data.

**Transition HDMAQ0:HDMAQ0:** When BSY is set to one and DMARQ is negated, the host shall make a transition to the HDMAQ0: Check_Status state.

**Transition HDMAQ0:HDMAQ1:** When BSY is cleared to zero, DRQ is set to one, and DMARQ is asserted, then the host shall set up the DMA engine and then make a transition to the HDMAQ1: Transfer_Data state.

**Transition HDMAQ0:HI0:** When BSY is cleared to zero, DRQ is cleared to zero, REL is cleared to zero, SERV is cleared to zero, and the device queue is empty, then the command is completed and the host shall make a transition to the HI0: Host_Idle state (see Figure 41). If an error is reported, the host shall perform appropriate error recovery.

**Transition HDMAQ0a:HIO0:** When BSY is cleared to zero, DRQ is cleared to zero, REL is cleared to zero, SERV is cleared to zero, nIEN is cleared to zero, and the device has a queue of released commands, then the command is completed and the host shall make a transition to the HIO0: Command completed state (see Figure 42). If an error is reported, the host shall perform appropriate error recovery.

**Transition HDMAQ0a:HIO3:** When BSY is cleared to zero, DRQ is cleared to zero, REL is cleared to zero, SERV is cleared to zero, nIEN is set to one, and the device has a queue of released commands, then the command is completed and the host shall make a transition to the HIO3: Command completed state (see Figure 42). If an error is reported, the host shall perform appropriate error recovery.

**Transition HDMAQ0b:HIO0:** When BSY is cleared to zero, DRQ is cleared to zero, REL is set to one, SERV is cleared to zero, and nIEN is cleared to zero, then the host shall make a transition to the HIO0: INTRQ_wait_A state (see Figure 42). The bus has been released.

**Transition HDMAQ0b:HIO3:** When BSY is cleared to zero, DRQ is cleared to zero, REL is set to one, SERV is cleared to zero, and nIEN is set to one, then the host shall make a transition to the HIO3: Check_status_A state (see Figure 42). The bus has been released.

**Transition HDMAQ0:HIO5:** When BSY is cleared to zero, DRQ is cleared to zero, and SERV is set to one, then the host shall make a transition to the HIO5: Write_SERVICE state (see Figure 42). The command is completed or the bus has been released, and another queued command is ready for service. If an error is reported, the host shall perform appropriate error recovery.

**HDMAQ1: Transfer_Data State:** This state is entered when BSY is cleared to zero, DRQ is set to one, and DMARQ is asserted.

When in this state, the host shall read or write the device Data port to transfer data. If the bus has been released, the host shall read the Tag in the Sector Count register to determine the queued command to be executed and initialize the DMA channel.

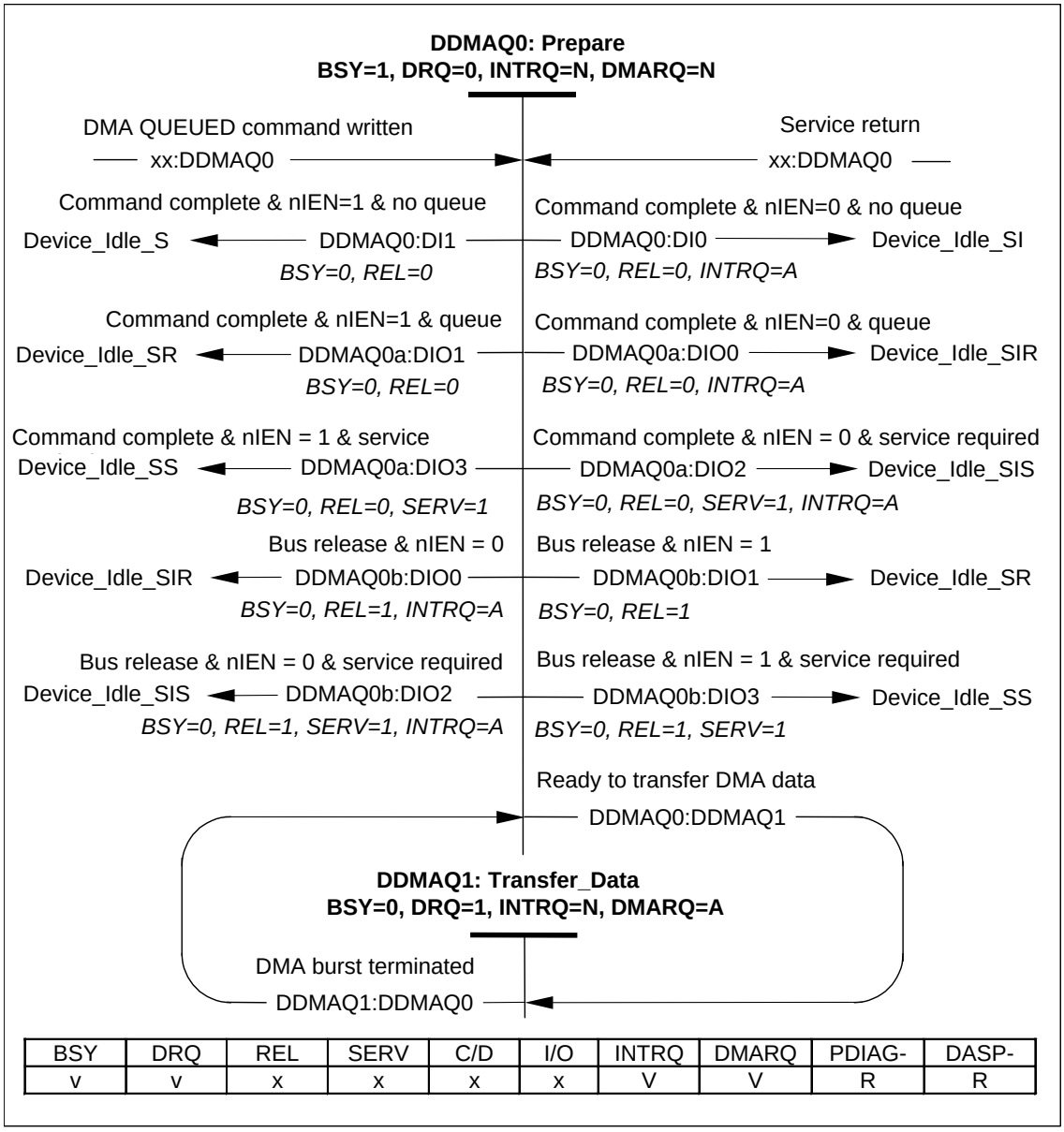
**Transition HDMAQ1:HDMAQ0:** The host shall make a transition to the HDMAQ0: Check_Status state when 1) all data for the request has been transferred and nIEN is set to one, or 2) the DMA burst is terminated and all data for the request has not been transferred.

**Transition HDMAQ1:HDMAQ2:** When all data for the request has been transferred and nIEN is cleared to zero, then the host shall make a transition to the HDMAQ2: INTRQ_wait state.

**HDMAQ2: INTRQ_Wait State:** This state is entered when the command has completed, and nIEN is cleared to zero.

When in this state, the host shall wait for INTRQ to be asserted.

**Transition HDMAQ2:HDMAQ0:** When INTRQ is asserted, the host shall make a transition to the HDMAQ0: Check_Status state.




| BSY | DRQ | REL | SERV | C/D | I/O | INTRQ | DMARQ | PDIAG- | DASP- |
|---|---|---|---|---|---|---|---|---|---|
| v | v | x | x | x | x | V | V | R | R |

**Figure 58 – Device DMA QUEUED command state diagram**

**DDMAQ0: Prepare State:** This state is entered when the device has a READ/WRITE DMA QUEUED or SERVICE command written to the Command register, when the data has been transferred, or when the command has completed.

When in this state, device shall set BSY to one, clear DRQ to zero, and negate INTRQ. If the command is a queued command, the device shall verify that the Tag is valid. If commands are queued, the Tag for the command to be serviced shall be placed into the Sector Count register.

**Transition DDMAQ0:DDMAQ1:** When the device is ready to transfer the data for a command, then the device shall make a transition to the DDMAQ1: Transfer_Data state.

**Transition DDMAQ0:DI0:** When the command has completed or an error occurs that causes the command to abort, the device has no other command released, and nIEN is cleared to zero, then the device shall set the interrupt pending, set appropriate error bits, clear BSY to zero, assert INTRQ, and make a transition to the DI0: Device_Idle_SI state (see Figure 43).

**Transition DDMAQ0:DI1:** When the command has completed or an error occurs that causes the command to abort, the device has no other command released, and nIEN is set to one, then the device shall set appropriate error bits, clear BSY to zero, assert INTRQ, and make a transition to the DI1: Device_Idle_S state (see Figure 43).

**Transition DDMAQ0a:DIO0:** When the command has completed or an error occurs that causes the command to abort, the device has another command released but not ready for service, and nIEN is cleared to zero, then the device shall set the interrupt pending, set appropriate error bits, clear BSY to zero, assert INTRQ, and make a transition to the DIO0: Device_Idle_SIR state (see Figure 44).

**Transition DDMAQ0a:DIO1:** When the command has completed or an error occurs that causes the command to abort, the device has another command released but not ready for service, and nIEN is set to one, then the device shall, set appropriate error bits, clear BSY to zero, and make a transition to the DIO1: Device_Idle_SR state (see Figure 44).

**Transition DDMAQ0a:DIO2:** When the command has completed or an error occurs that causes the command to abort, the device has another command ready for service, and nIEN is cleared to zero, then the device shall set the interrupt pending, set appropriate error bits, set SERV to one, clear BSY to zero, assert INTRQ, and make a transition to the DIO2: Device_Idle_SIS state (see Figure 44).

**Transition DDMAQ0a:DIO3:** When the command has completed or an error occurs that causes the command to abort, the device has another command ready for service, and nIEN is set to one, then the device shall set appropriate error bits, set SERV to one, clear BSY to zero, and make a transition to the DIO3: Device_Idle_SS state (see Figure 44).

**Transition DDMAQ0b:DIO0:** When the bus is released and nIEN is cleared to zero, then the device shall set the interrupt pending, set appropriate error bits, set REL to one, clear BSY to zero, assert INTRQ, and make a transition to the DIO0: Device_Idle_SIR state (see Figure 44).

**Transition DDMAQ0b:DIO1:** When the bus is released and nIEN is set to one, then the device shall, set appropriate error bits, set REL to one, clear BSY to zero, and make a transition to the DIO1: Device_Idle_SR state (see Figure 44).

**Transition DDMAQ0b:DIO2:** When the bus is released, the device has another command ready for service, and nIEN is cleared to zero, then the device shall set the interrupt pending, set appropriate error bits, set REL to one, set SERV to one, clear BSY to zero, assert INTRQ, and make a transition to the DIO2: Device_Idle_SIS state (see Figure 44).

**Transition DDMAQ0b:DIO3:** When the bus is released, the device has another command ready for service, and nIEN is set to one, then the device shall set appropriate error bits, set REL to one, set SERV to one, clear BSY to zero, and make a transition to the DIO3: Device_Idle_SS state (see Figure 44).

**DDMAQ1: Data_Transfer State:** This state is entered when the device is ready to transfer DMA data.

When in this state, BSY is cleared to zero, DRQ is set to one, INTRQ is negated, DMARQ is asserted, and data is transferred as described in Multiword DMA timing or Ultra DMA protocol.

**Transition DDMAQ1:DDMAQ0:** When the DMA burst has been terminated, then the device shall make a transition to the DDMAQ0: Prepare state. All of the data for the command may not have been transferred.

## 11.10 EXECUTE DEVICE DIAGNOSTIC command protocol

This class includes:

- EXECUTE DEVICE DIAGNOSTIC

If the host asserts RESET- before devices have completed executing their EXECUTE DEVICE DIAGNOSTIC protocol, then the devices shall start executing the power-on or hardware reset protocol from the beginning.

If the host sets SRST to one in the Device Control register before the devices have completed execution of their EXECUTE DEVICE DIAGNOSTIC protocol, then the devices shall start executing their software reset protocol from the beginning.

Figure 59 and the text following the figure describe the EXECUTE DEVICE DIAGNOSTIC protocol for the host. Figure 60 and the text following the figure describe the EXECUTE DEVICE DIAGNOSTIC protocol for Device 0. Figure 61 and the text following the figure describe the EXECUTE DEVICE DIAGNOSTIC protocol for Device 1.

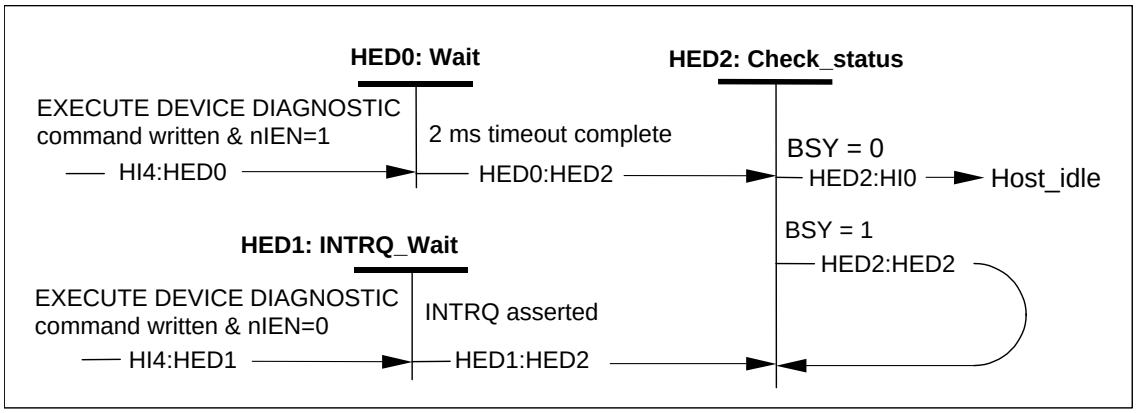



**Figure 59 – Host EXECUTE DEVICE DIAGNOSTIC state diagram**

**HED0: Wait State:** This state is entered when the host has written the EXECUTE DEVICE DIAGNOSTIC command to the devices and nIEN is set to one.

The host shall remain in this state for at least 2 ms.

**Transition HED0:HED1:** When at least 2 ms has elapsed since the command was written, the host shall make a transition to the HED1: Check_status state.

**HED1: INTRQ_wait:** This state is entered when the host has written the EXECUTE DEVICE DIAGNOSTIC command to the devices and nIEN is cleared to zero.

When in this state the host shall wait for INTRQ to be asserted.

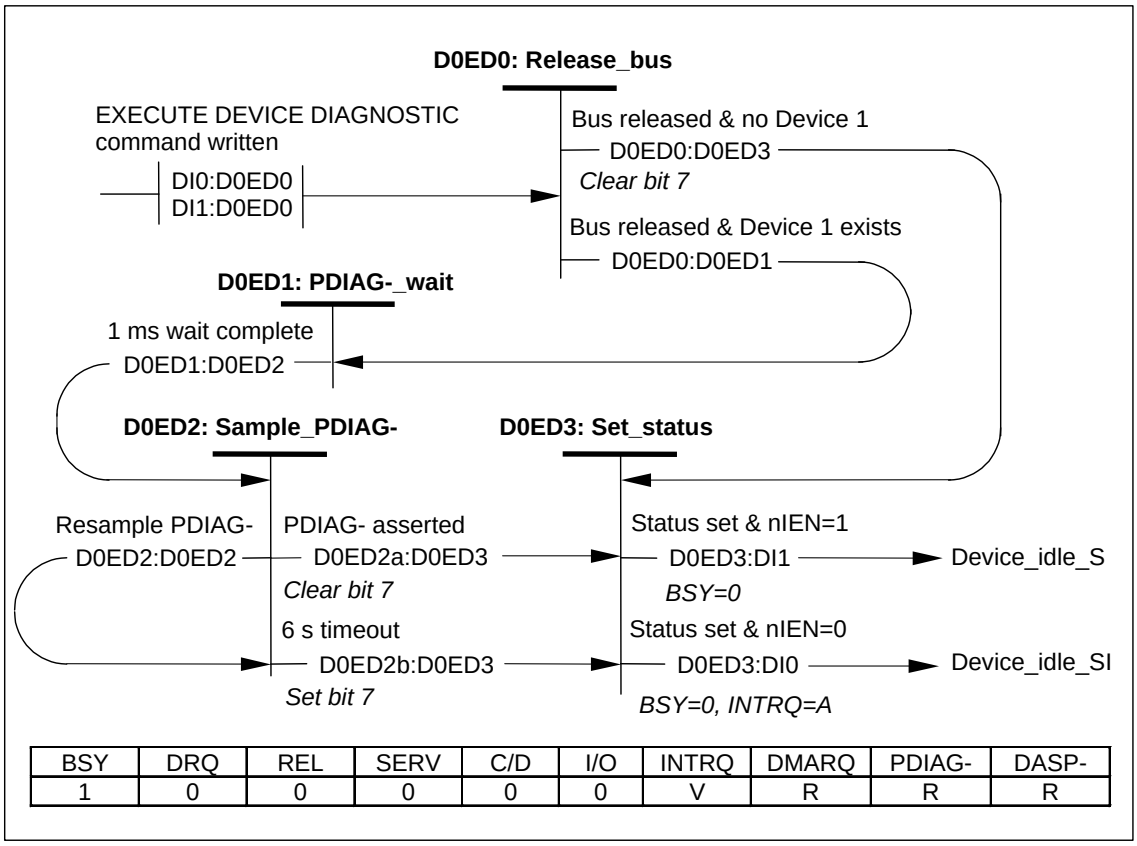
**Transition HED1:HED2:** When INTRQ is asserted, the host shall make a transition to the HED2: Check_status state.

**HED2: Check_status State:** This state is entered when at least 2 ms since the command was written or INTRQ has been asserted.

When in this state, the host shall read the Status or Alternate Status register.

**Transition HED2:HED2:** When BSY is set to one, the host shall make a transition to the HED1: Check_status state.

**Transition HED2:HI0:** When BSY is cleared to zero, the host shall check the results of the command (see clause 5) and make a transition to the HI0: Host_idle state (see Figure 41).




| BSY | DRQ | REL | SERV | C/D | I/O | INTRQ | DMARQ | PDIAG- | DASP- |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | V | R | R | R |

**Figure 60 – Device 0 EXECUTE DEVICE DIAGNOSTIC state diagram**

**D0ED0: Release_bus State:** This state is entered when the EXECUTE DEVICE DIAGNOSTIC command has been written.

When in this state, the device shall release PDIAG-, INTRQ, IORDY, DMARQ, and DD(15:0) and shall set BSY to one within 400 ns after entering this state.

The device should begin performing the self-diagnostic testing.

**Transition D0ED0:D0ED1:** When the bus has been released, BSY set to one, and the assertion of DASP- by Device 1 was detected during the most recent power-on or hardware reset, then the device shall make a transition to the D0ED1: PDIAG-_wait state.

**Transition D0ED0:D0ED3:** When the bus has been released, BSY set to one, and the assertion of DASP- by Device 1 was not detected during the most recent power-on or hardware reset, then the device shall clear bit 7 in the Error register and make a transition to the D0ED3: Set_status state.

**D0ED1: PDIAG-_wait State:** This state is entered when the bus has been released, BSY set to one, and Device 1 exists.

The device shall remain in this state until least 1 ms has elapsed since the command was written and shall clear the DEV bit in the Device register to zero within 1 ms.

**Transition D0ED1:D0ED2:** When at least 1 ms has elapsed since the command was written, the device shall make a transition to the D0ED2: Sample_PDIAG- state.

**D0ED2: Sample_PDIAG- State:** This state is entered when at least 1 ms has elapsed since the command was written.

When in this state, the device shall sample the PDIAG- signal.

**Transition D0ED2:D0ED3:** When the sample indicates that PDIAG- is asserted, the device shall clear bit 7 in the Error register and make a transition to the D0ED3: Set_status state.

**Transition D0ED2:D0ED2:** When the sample indicates that PDIAG- is not asserted and less than 6 s have elapsed since the command was written, then the device shall make a transition to the D0ED2: Sample_PDIAG- state.

**Transition D0ED2:D0ED3:** When the sample indicates that DASP- is not asserted and 6 s have elapsed since the command was written, then the device shall set bit 7 in the Error register and make a transition to the D0ED3: Set_status state.

**D0ED3: Set_status State:** This state is entered when Bit 7 in the Error register has been set or cleared.

When in this state, the device shall clear the DEV bit in the Device register to zero within 1 ms. The device shall complete the self-diagnostic testing begun in the Release bus state if not already completed.

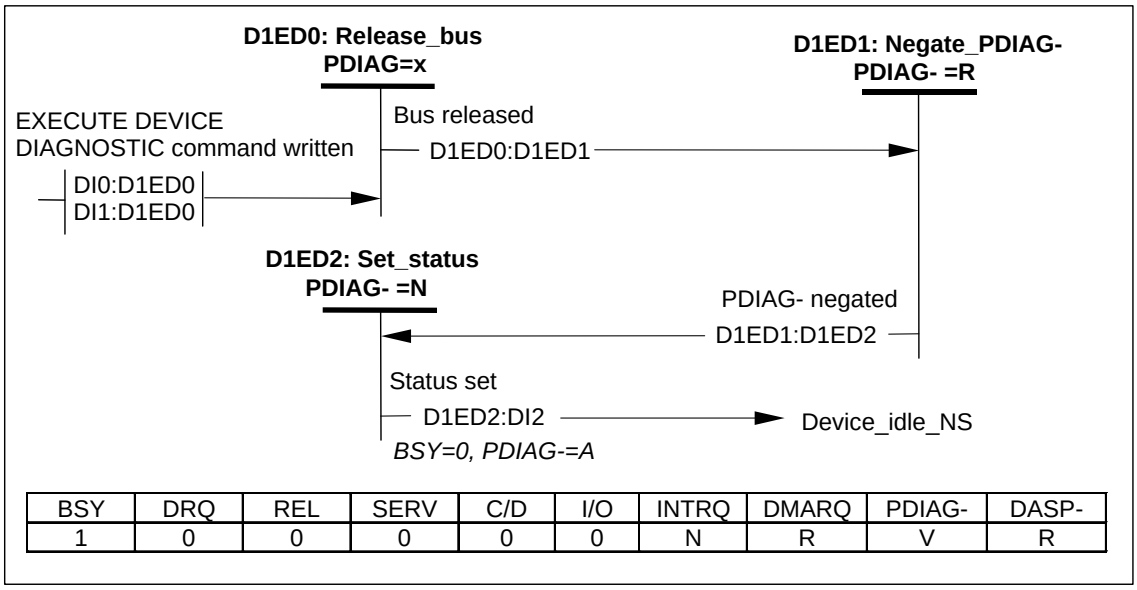
Results of the EXECUTE DEVICE DIAGNOSTICS self-diagnostic testing shall be placed in bits (6:0) of the Error register (see clause 6). The device shall set the signature values (see clause 5). The contents of the Features register is undefined.

If the device does not implement the PACKET command feature set, the device shall clear bits 3, 2, and 0 in the Status register to zero.

If the device implements the PACKET command feature set, the device shall clear bits 6, 5, 4, 3, 2, and 0 in the Status register to zero. The device shall return the operating modes to their specified initial conditions. MODE SELECT conditions shall be restored to their last saved values if saved values have been established. MODE SELECT conditions for which no values have been saved shall be returned to their default values.

**Transition D0ED3:DI1:** When hardware initialization and self-diagnostic testing is completed, the status has been set, and nIEN is set to one, then the device shall clear BSY to zero, and make a transition to the DI1: Device_idle_S state (see Figure 43).

**Transition D0ED3:DI0:** When hardware initialization and self-diagnostic testing is completed, the status has been set, and nIEN is cleared to zero, then the device shall clear BSY to zero, assert INTRQ, and make a transition to the DI0: Device_idle_SI state (see Figure 43).




| BSY | DRQ | REL | SERV | C/D | I/O | INTRQ | DMARQ | PDIAG- | DASP- |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | N | R | V | R |

**Figure 61 – Device 1 EXECUTE DEVICE DIAGNOSTIC command state diagram**

**D1ED0: Release_bus State:** This state is entered when the EXECUTE DEVICE DIAGNOSTIC command is written.

When in this state, the device shall release INTRQ, IORDY, DMARQ, and DD(15:0) within 400 ns after entering this state. The device shall set BSY to one within 400 ns after entering this state.

The device should begin performing the self-diagnostic testing.

**Transition D1ED0:D1ED1:** When the bus has been released and BSY set to one, then the device shall make a transition to the D1ED1: Negate_PDIAG- state.

**D1ED1: Negate_PDIAG- State:** This state is entered when the bus has been released and BSY set to one.

When in this state, the device shall negate PDIAG- and clear the DEV bit in the Device register within less than 1 ms of the receipt of the EXECUTE DEVICE DIAGNOSTIC command.

**Transition D1ED1:D1ED2:** When PDIAG- has been negated, the device shall make a transition to the D1ED2: Set_status state.

**D1ED2: Set_status State:** This state is entered when the device has negated PDIAG-.

When in this state the device shall complete the hardware initialization and self-diagnostic testing begun in the Release bus state if not already completed. Results of the EXECUTE DEVICE DIAGNOSTICS shall be placed in the Error register (see clause 6). If the device passed the self-diagnostics, the device shall assert PDIAG-.

The device shall set the signature values (see clause 5). The effect on the Features register is undefined.

If the device does not implement the PACKET command feature set, the device shall clear bits 3, 2, and 0 in the Status register to zero.

If the device implements the PACKET command feature set, the device shall clear bits 6, 5, 4, 3, 2, and 0 in the Status register to zero. The device shall return the operating modes to their specified initial conditions. MODE SELECT conditions shall be restored to their last saved values if saved values have been

established. MODE SELECT conditions for which no values have been saved shall be returned to their default values.

All requirements for this state shall be completed within 5 s from the writing of the command.

**Transition D1ED2:DI2:** When hardware initialization and self-diagnostic testing is completed and the status has been set, then the device shall clear BSY to zero, assert PDIAG- if diagnostics were passed, and make a transition to the DI2: Device_idle_NS state (see Figure 43).

## 11.11 DEVICE RESET command protocol

This class includes:

– DEVICE RESET

If the host asserts RESET- before the device has completed executing a DEVICE RESET command, then the device shall start executing the hardware reset protocol from the begining. If the host sets the SRST bit to one in the Device Control register before the device has completed executing a DEVICE RESET command, the device shall start executing the software reset protocol from the beginning.

The host should not issue a DEVICE RESET command while a DEVICE RESET command is in progress. If the host issues a DEVICE RESET command while a DEVICE RESET command is in progress, the results are indeterminate.

Figure 62 and the text following the figure describe the DEVICE RESET command protocol for the host. Figure 63 and the text following the figure describe the DEVICE RESET command protocol for the device.

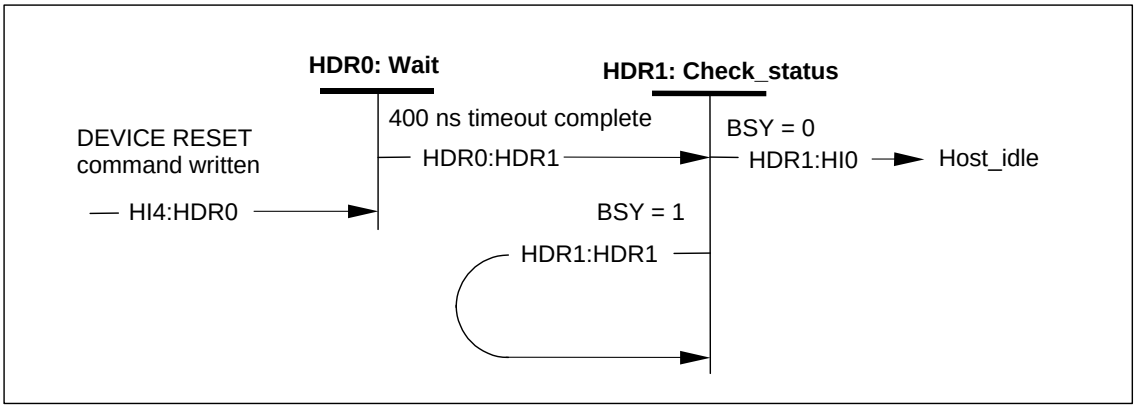



**Figure 62 – Host DEVICE RESET command state diagram**

**HDR0: Wait State:** This state is entered when the host has written the DEVICE RESET command to the device.

The host shall remain in this state for at least 400 ns.

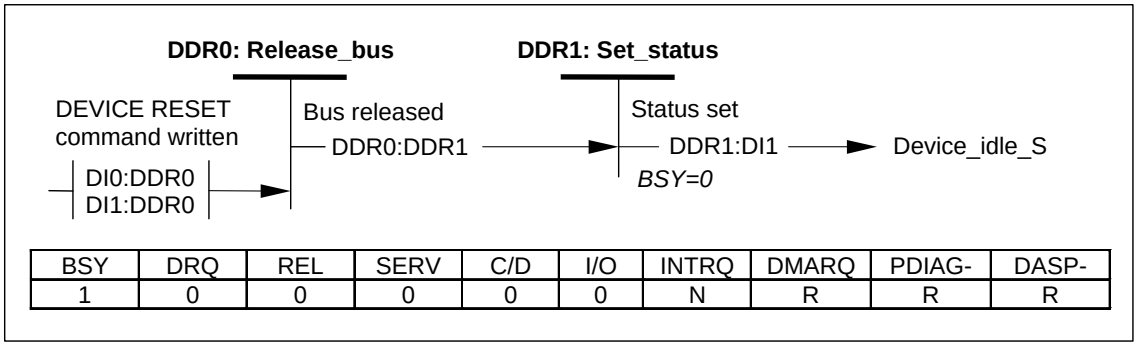
**Transition HDR0:HDR1:** When at least 400 ns has elapsed since the command was written, the host shall make a transition to the HDR1: Check_status state.

**HDR1: Check_status State:** This state is entered when at least 400 ns has elapsed since the command was written.

When in this state the host shall read the Status register.

**Transition HDR1:HDR1:** When BSY is set to one, the host shall make a transition to the HDR1: Check_status state.

**Transition HDR1:HI0:** When BSY is cleared to zero, the host shall make a transition to the HI0: Host_idle state (see Figure 41). If status indicates that an error has occurred, the host shall take appropriate action.




| BSY | DRQ | REL | SERV | C/D | I/O | INTRQ | DMARQ | PDIAG- | DASP- |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | N | R | R | R |

**Figure 63 – Device DEVICE RESET command state diagram**

**DDR0: Release_bus State:** This state is entered when the DEVICE RESET command is written.

When in this state, the device shall release INTRQ, IORDY, DMARQ, and DD(15:0) within 400 ns after entering this state. The device shall set BSY to one within 400 ns after entering this state.

**Transition DDR0:DDR1:** When the bus has been released and BSY set to one, the device shall make a transition to the DDR1: Set_status state.

**DDR1: Set_status State:** This state is entered when the device has released the bus and set BSY to one.

When in this state the device should stop execution of any uncompleted command. The device should end background activity (e.g., immediate commands, see MMC and MMC-2).

The device should not revert to the default condition. If the device reverts to the default condition, the device shall report an exception condition by setting CHK to one in the Status register. MODE SELECT conditions shall not be altered.

The device shall set the signature values (see clause 5). The content of the Features register is undefined.

The device shall clear bit 7 in the ERROR register to zero. The device shall clear bits 6, 5, 4, 3, 2, and 0 in the Status register to zero.

**Transition DDR1:DI1:** When the status has been set, the device shall clear BSY to zero and make a transition to the DI1: Device_idle_S state (see Figure 43).

## 11.12 Ultra DMA data-in commands

### 11.12.1 Initiating an Ultra DMA data-in burst

The following steps shall occur in the order they are listed unless otherwise specified. Timing requirements are shown in 12.2.4 and12.2.4.1.

a) The host shall keep DMACK- in the negated state before an Ultra DMA burst is initiated.
b) The device shall assert DMARQ to initiate an Ultra DMA burst when DMACK- is negated. After assertion of DMARQ the device shall not negate DMARQ until after the first negation of DSTROBE.
c) Steps (c), (d), and (e) may occur in any order or at the same time. The host shall assert STOP.
d) The host shall negate HDMARDY-.
e) The host shall negate CS0-, CS1-, DA2, DA1, and DA0. The host shall keep CS0-, CS1-, DA2, DA1, and DA0 negated until after negating DMACK- at the end of the burst.

f) Steps (c), (d), and (e) shall have occurred at least $t_{ACK}$ before the host asserts DMACK-. The host shall keep DMACK- asserted until the end of an Ultra DMA burst.
g) The host shall release DD(15:0) within $t_{AZ}$ after asserting DMACK-.
h) The device may assert DSTROBE $t_{ZIORDY}$ after the host has asserted DMACK-. Once the device has driven DSTROBE the device shall not release DSTROBE until after the host has negated DMACK- at the end of an Ultra DMA burst.
i) The host shall negate STOP and assert HDMARDY- within $t_{ENV}$ after asserting DMACK-. After negating STOP and asserting HDMARDY-, the host shall not change the state of either signal until after receiving the first negation of DSTROBE from the device (i.e., after the first data word has been received).
j) The device shall drive DD(15:0) no sooner than $t_{ZAD}$ after the host has asserted DMACK-, negated STOP, and asserted HDMARDY-.
k) The device shall drive the first word of the data transfer onto DD(15:0). This step may occur when the device first drives DD(15:0) in step (j).
l) To transfer the first word of data the device shall negate DSTROBE within $t_{FS}$ after the host has negated STOP and asserted HDMARDY-. The device shall negate DSTROBE no sooner than $t_{DVS}$ after driving the first word of data onto DD(15:0).

#### 11.12.2 The data-in transfer

The following steps shall occur in the order they are listed unless otherwise specified. Timing requirements are shown in 12.2.4 and 12.2.4.2.

a) The device shall drive a data word onto DD(15:0).
b) The device shall generate a DSTROBE edge to latch the new word no sooner than $t_{DVS}$ after changing the state of DD(15:0). The device shall generate a DSTROBE edge no more frequently than $t_{CYC}$ for the selected Ultra DMA mode. The device shall not generate two rising or two falling DSTROBE edges more frequently than $t_{2cyc}$ for the selected Ultra DMA mode.
c) The device shall not change the state of DD(15:0) until at least $t_{DVH}$ after generating a DSTROBE edge to latch the data.
d) The device shall repeat steps (a), (b), and (c) until the Ultra DMA burst is paused or terminated by the device or host.

#### 11.12.3 Pausing an Ultra DMA data-in burst

The following steps shall occur in the order they are listed unless otherwise specified. Timing requirements are shown in 12.2.4 and 12.2.4.3.

##### 11.12.3.1 Device pausing an Ultra DMA data-in burst

a) The device shall not pause an Ultra DMA burst until at least one data word of an Ultra DMA burst has been transferred.
b) The device shall pause an Ultra DMA burst by not generating additional DSTROBE edges. If the host is ready to terminate the Ultra DMA burst (see 11.12.4.2).
c) The device shall resume an Ultra DMA burst by generating a DSTROBE edge.

##### 11.12.3.2 Host pausing an Ultra DMA data-in burst

a) The host shall not pause an Ultra DMA burst until at least one data word of an Ultra DMA burst has been transferred.
b) The host shall pause an Ultra DMA burst by negating HDMARDY-.
c) The device shall stop generating DSTROBE edges within $t_{RFS}$ of the host negating HDMARDY-.
d) When operating in Ultra DMA modes 2, 1, or 0 the host shall be prepared to receive zero, one, or two additional data words after negating HDMARDY-. While operating in Ultra DMA modes 6, 5, 4, or 3 the host shall be prepared to receive zero, one, two, or three additional data words after negating HDMARDY-. The additional data words are a result of cable round trip delay and $t_{RFS}$ timing for the device.
e) The host shall resume an Ultra DMA burst by asserting HDMARDY-.

### 11.12.4 Terminating an Ultra DMA data-in burst

#### 11.12.4.1 Device terminating an Ultra DMA data-in burst

Burst termination is completed when the termination protocol has been executed and DMACK- negated.

The device shall terminate an Ultra DMA burst before command completion.

The following steps shall occur in the order they are listed unless otherwise specified. Timing requirements are shown in 12.2.4 and 12.2.4.4.

a) The device shall initiate termination of an Ultra DMA burst by not generating additional DSTROBE edges.
b) The device shall negate DMARQ no sooner than $t_{SS}$ after generating the last DSTROBE edge. The device shall not assert DMARQ again until after DMACK- has been negated.
c) The device shall release DD(15:0) no later than $t_{AZ}$ after negating DMARQ.
d) The host shall assert STOP within $t_{LI}$ after the device has negated DMARQ. The host shall not negate STOP again until after the Ultra DMA burst is terminated.
e) The host shall negate HDMARDY- within $t_{LI}$ after the device has negated DMARQ. The host shall continue to negate HDMARDY- until the Ultra DMA burst is terminated. Steps (d) and (e) may occur at the same time.
f) The host shall drive DD(15:0) no sooner than $t_{ZAH}$ after the device has negated DMARQ. For this step, the host may first drive DD(15:0) with the result of the host CRC calculation (see 11.14);
g) If DSTROBE is negated, the device shall assert DSTROBE within $t_{LI}$ after the host has asserted STOP. No data shall be transferred during this assertion. The host shall ignore this transition on DSTROBE. DSTROBE shall remain asserted until the Ultra DMA burst is terminated.
h) If the host has not placed the result of the host CRC calculation on DD(15:0) since first driving DD(15:0) during (f), the host shall place the result of the host CRC calculation on DD(15:0) (see 11.14).
i) The host shall negate DMACK- no sooner than $t_{MLI}$ after the device has asserted DSTROBE and negated DMARQ and the host has asserted STOP and negated HDMARDY-, and no sooner than $t_{DVS}$ after the host places the result of the host CRC calculation on DD(15:0).
j) The device shall latch the host's CRC data from DD(15:0) on the negating edge of DMACK-.
k) The device shall compare the CRC data received from the host with the results of the device CRC calculation. If a miscompare error occurs during one or more Ultra DMA bursts for any one command, at the end of the command the device shall report the first error that occurred (see 11.14).
l) The device shall release DSTROBE within $t_{IORDYZ}$ after the host negates DMACK-.
m) The host shall not negate STOP nor assert HDMARDY- until at least $t_{ACK}$ after negating DMACK-.
n) The host shall not assert DIOR-, CS0-, CS1-, DA2, DA1, or DA0 until at least $t_{ACK}$ after negating DMACK.

#### 11.12.4.2 Host terminating an Ultra DMA data-in burst

The following steps shall occur in the order they are listed unless otherwise specified. Timing requirements are shown in 12.2.4 and 12.2.4.5.

a) The host shall not initiate Ultra DMA burst termination until at least one data word of an Ultra DMA burst has been transferred.
b) The host shall initiate Ultra DMA burst termination by negating HDMARDY-. The host shall continue to negate HDMARDY- until the Ultra DMA burst is terminated.
c) The device shall stop generating DSTROBE edges within $t_{RFS}$ of the host negating HDMARDY-.
d) When operating in Ultra DMA modes 2, 1, or 0 the host shall be prepared to receive zero, one or two additional data words after negating HDMARDY-. While operating in Ultra DMA modes 6, 5, 4, or 3 the host shall be prepared to receive zero, one, two, or three additional data words after negating HDMARDY-. The additional data words are a result of cable round trip delay and $t_{RFS}$ timing for the device.
e) The host shall assert STOP no sooner than $t_{RP}$ after negating HDMARDY-. The host shall not negate STOP again until after the Ultra DMA burst is terminated.
f) The device shall negate DMARQ within $t_{LI}$ after the host has asserted STOP. The device shall not assert DMARQ again until after the Ultra DMA burst is terminated.

g) If DSTROBE is negated, the device shall assert DSTROBE within $t_{LI}$ after the host has asserted STOP. No data shall be transferred during this assertion. The host shall ignore this transition on DSTROBE. DSTROBE shall remain asserted until the Ultra DMA burst is terminated.
h) The device shall release DD(15:0) no later than $t_{AZ}$ after negating DMARQ.
i) The host shall drive DD(15:0) no sooner than $t_{ZAH}$ after the device has negated DMARQ. For this step, the host may first drive DD(15:0) with the result of the host CRC calculation (see 11.14).
j) If the host has not placed the result of the host CRC calculation on DD(15:0) since first driving DD(15:0) during (9), the host shall place the result of the host CRC calculation on DD(15:0) (see 11.14).
k) The host shall negate DMACK- no sooner than $t_{MLI}$ after the device has asserted DSTROBE and negated DMARQ and the host has asserted STOP and negated HDMARDY-, and no sooner than $t_{DVS}$ after the host places the result of the host CRC calculation on DD(15:0).
l) The device shall latch the host's CRC data from DD(15:0) on the negating edge of DMACK-.
m) The device shall compare the CRC data received from the host with the results of the device CRC calculation. If a miscompare error occurs during one or more Ultra DMA burst for any one command, at the end of the command, the device shall report the first error that occurred (see 11.14) .
n) The device shall release DSTROBE within $t_{IORDYZ}$ after the host negates DMACK-.
o) The host shall neither negate STOP nor assert HDMARDY- until at least $t_{ACK}$ after the host has negated DMACK-.
p) The host shall not assert DIOR-, CS0-, CS1-, DA2, DA1, or DA0 until at least $t_{ACK}$ after negating DMACK.

## 11.13 Ultra DMA data-out commands

### 11.13.1 Initiating an Ultra DMA data-out burst

The following steps shall occur in the order they are listed unless otherwise specified. Timing requirements are shown in 12.2.4 and 12.2.4.6.

a) The host shall keep DMACK- in the negated state before an Ultra DMA burst is initiated.
b) The device shall assert DMARQ to initiate an Ultra DMA burst when DMACK- is negated.
c) Steps (c), (d), and (e) may occur in any order or at the same time. The host shall assert STOP.
d) The host shall assert HSTROBE.
e) The host shall negate CS0-, CS1-, DA2, DA1, and DA0. The host shall keep CS0-, CS1-, DA2, DA1, and DA0 negated until after negating DMACK- at the end of the burst.
f) Steps (c), (d), and (e) shall have occurred at least $t_{ACK}$ before the host asserts DMACK-. The host shall keep DMACK- asserted until the end of an Ultra DMA burst.
g) The device may negate DDMARDY- $t_{ZIORDY}$ after the host has asserted DMACK-. Once the device has negated DDMARDY-, the device shall not release DDMARDY- until after the host has negated DMACK- at the end of an Ultra DMA burst.
h) The host shall negate STOP within $t_{ENV}$ after asserting DMACK-. The host shall not assert STOP until after the first negation of HSTROBE.
i) The device shall assert DDMARDY- within $t_{LI}$ after the host has negated STOP. After asserting DMARQ and DDMARDY- the device shall not negate either signal until after the first negation of HSTROBE by the host.
j) The host shall drive the first word of the data transfer onto DD(15:0). This step may occur any time during Ultra DMA burst initiation.
k) To transfer the first word of data: the host shall negate HSTROBE no sooner than $t_{UI}$ after the device has asserted DDMARDY-. The host shall negate HSTROBE no sooner than $t_{DVS}$ after the driving the first word of data onto DD(15:0).

### 11.13.2 The data-out transfer

The following steps shall occur in the order they are listed unless otherwise specified. Timing requirements are shown in 12.2.4 and 12.2.4.7.

a) The host shall drive a data word onto DD(15:0).
b) The host shall generate an HSTROBE edge to latch the new word no sooner than $t_{DVS}$ after changing the state of DD(15:0). The host shall generate an HSTROBE edge no more frequently than $t_{CYC}$ for the

selected Ultra DMA mode. The host shall not generate two rising or falling HSTROBE edges more frequently than $t_{2cyc}$ for the selected Ultra DMA mode.

c) The host shall not change the state of DD(15:0) until at least $t_{DVH}$ after generating an HSTROBE edge to latch the data.
d) The host shall repeat steps (a), (b), and (c) until the Ultra DMA burst is paused or terminated by the device or host.

### 11.13.3 Pausing an Ultra DMA data-out burst

The following steps shall occur in the order they are listed unless otherwise specified. Timing requirements are shown in 12.2.4 and 12.2.4.8.

#### 11.13.3.1 Host pausing an Ultra DMA data-out burst

a) The host shall not pause an Ultra DMA burst until at least one data word of an Ultra DMA burst has been transferred.
b) The host shall pause an Ultra DMA burst by not generating an HSTROBE edge . If the host is ready to terminate the Ultra DMA burst (see 11.13.4.1).
c) The host shall resume an Ultra DMA burst by generating an HSTROBE edge.

#### 11.13.3.2 Device pausing an Ultra DMA data-out burst

a) The device shall not pause an Ultra DMA burst until at least one data word of an Ultra DMA burst has been transferred.
b) The device shall pause an Ultra DMA burst by negating DDMARDY-.
c) The host shall stop generating HSTROBE edges within $t_{RFS}$ of the device negating DDMARDY-.
d) When operating in Ultra DMA modes 2, 1, or 0 the device shall be prepared to receive zero, one or two additional data words after negating DDMARDY-. While operating in Ultra DMA modes 6, 5, 4 or 3 the device shall be prepared to receive zero, one, two or three additional data words after negating DDMARDY-. The additional data words are a result of cable round trip delay and $t_{RFS}$ timing for the host.
e) The device shall resume an Ultra DMA burst by asserting DDMARDY-.

### 11.13.4 Terminating an Ultra DMA data-out burst

#### 11.13.4.1 Host terminating an Ultra DMA data-out burst

The following steps shall occur in the order they are listed unless otherwise specified. Timing requirements are shown in 12.2.4 and 12.2.4.9.

a) The host shall initiate termination of an Ultra DMA burst by not generating additional HSTROBE edges.
b) The host shall assert STOP no sooner than $t_{SS}$ after the last generated an HSTROBE edge. The host shall not negate STOP again until after the Ultra DMA burst is terminated.
c) The device shall negate DMARQ within $t_{LI}$ after the host asserts STOP. The device shall not assert DMARQ again until after the Ultra DMA burst is terminated.
d) The device shall negate DDMARDY- within $t_{LI}$ after the host has negated STOP. The device shall not assert DDMARDY- again until after the Ultra DMA burst termination is complete.
e) If HSTROBE is negated, the host shall assert HSTROBE within $t_{LI}$ after the device has negated DMARQ. No data shall be transferred during this assertion. The device shall ignore this transition on HSTROBE. HSTROBE shall remain asserted until the Ultra DMA burst is terminated.
f) The host shall place the result of the host CRC calculation on DD(15:0) (see 11.14).
g) The host shall negate DMACK- no sooner than $t_{MLI}$ after the host has asserted HSTROBE and STOP and the device has negated DMARQ and DDMARDY-, and no sooner than $t_{DVS}$ after placing the result of the host CRC calculation on DD(15:0).
h) The device shall latch the host's CRC data from DD(15:0) on the negating edge of DMACK-.
i) The device shall compare the CRC data received from the host with the results of the device CRC calculation. If a miscompare error occurs during one or more Ultra DMA bursts for any one command, at the end of the command, the device shall report the first error that occurred (see 11.14).
j) The device shall release DDMARDY- within $t_{IORDYZ}$ after the host has negated DMACK-.

k) The host shall neither negate STOP nor negate HSTROBE until at least $t_{ACK}$ after negating DMACK-.
l) The host shall not assert DIOW-, CS0-, CS1-, DA2, DA1, or DA0 until at least $t_{ACK}$ after negating DMACK.

#### 11.13.4.2 Device terminating an Ultra DMA data-out burst

Burst termination is completed when the termination protocol has been executed and DMACK- negated.

The device shall terminate an Ultra DMA burst before command completion.

The following steps shall occur in the order they are listed unless otherwise specified. Timing requirements are shown in 12.2.4 and 12.2.4.10.

a) The device shall not initiate Ultra DMA burst termination until at least one data word of an Ultra DMA burst has been transferred.
b) The device shall initiate Ultra DMA burst termination by negating DDMARDY-.
c) The host shall stop generating an HSTROBE edges within $t_{RFS}$ of the device negating DDMARDY-.
d) When operating in Ultra DMA modes 2, 1, or 0 the device shall be prepared to receive zero, one or two additional data words after negating DDMARDY-. While operating in Ultra DMA modes 6, 5, 4 or 3 the device shall be prepared to receive zero, one, two or three additional data words after negating DDMARDY-. The additional data words are a result of cable round trip delay and $t_{RFS}$ timing for the host.
e) The device shall negate DMARQ no sooner than $t_{RP}$ after negating DDMARDY-. The device shall not assert DMARQ again until after DMACK- is negated.
f) The host shall assert STOP within $t_{LI}$ after the device has negated DMARQ. The host shall not negate STOP again until after the Ultra DMA burst is terminated.
g) If HSTROBE is negated, the host shall assert HSTROBE within $t_{LI}$ after the device has negated DMARQ. No data shall be transferred during this assertion. The device shall ignore this transition of HSTROBE. HSTROBE shall remain asserted until the Ultra DMA burst is terminated.
h) The host shall place the result of the host CRC calculation on DD(15:0) (see 11.14).
i) The host shall negate DMACK- no sooner than $t_{MLI}$ after the host has asserted HSTROBE and STOP and the device has negated DMARQ and DDMARDY-, and no sooner than $t_{DVS}$ after placing the result of the host CRC calculation on DD(15:0).
j) The device shall latch the host's CRC data from DD(15:0) on the negating edge of DMACK-.
k) The device shall compare the CRC data received from the host with the results of the device CRC calculation. If a miscompare error occurs during one or more Ultra DMA bursts for any one command, at the end of the command, the device shall report the first error that occurred (see 11.14).
l) The device shall release DDMARDY- within $t_{IORDYZ}$ after the host has negated DMACK-.
m) The host shall neither negate STOP nor HSTROBE until at least $t_{ACK}$ after negating DMACK-.
n) The host shall not assert DIOW-, CS0-, CS1-, DA2, DA1, or DA0 until at least $t_{ACK}$ after negating DMACK.

## 11.14 Ultra DMA CRC rules

The following is a list of rules for calculating CRC, determining if a CRC error has occurred during an Ultra DMA burst, and reporting any error that occurs at the end of a command.

1) Both the host and the device shall have a 16-bit CRC calculation function.
2) Both the host and the device shall calculate a CRC value for each Ultra DMA burst.
3) The CRC function in the host and the device shall be initialized with a seed of 4ABAh at the beginning of an Ultra DMA burst before any data is transferred.
4) For each STROBE transition used for data transfer, both the host and the device shall calculate a new CRC value by applying the CRC polynomial to the current value of their individual CRC functions and the word being transferred. CRC is not calculated for the return of STROBE to the asserted state after the Ultra DMA burst termination request has been acknowledged.
5) At the end of any Ultra DMA burst the host shall send the results of the host CRC calculation function to the device on DD(15:0) with the negation of DMACK-.
6) The device shall then compare the CRC data from the host with the calculated value in its own CRC calculation function. If the two values do not match, the device shall save the error. A subsequent Ultra

DMA burst for the same command that does not have a CRC error shall not clear an error saved from a previous Ultra DMA burst in the same command. If a miscompare error occurs during one or more Ultra DMA bursts for any one command, the device shall report the first error that occurred. If the device detects that a CRC error has occurred before data transfer for the command is complete, the device may complete the transfer and report the error or abort the command and report the error.

7) For READ DMA, WRITE DMA, READ DMA QUEUED, or WRITE DMA QUEUED commands: When a CRC error is detected, the error shall be reported by setting both ICRC and ABRT (bit 7 and bit 2 in the Error register) to one. ICRC is defined as the Interface CRC Error bit. The host shall respond to this error by re-issuing the command.
8) For a REQUEST SENSE packet command (see SPC NCITS 301:1997 for definition of the REQUEST SENSE command): When a CRC error is detected during transmission of sense data the device shall complete the command and set CHK to one. The device shall report a Sense key of 0Bh (ABORTED COMMAND). The device shall preserve the original sense data that was being returned when the CRC error occurred. The device shall not report any additional sense data specific to the CRC error. The host device driver may retry the REQUEST SENSE command or may consider this an unrecoverable error and retry the command that caused the Check Condition.
9) For any packet command except a REQUEST SENSE command: If a CRC error is detected, the device shall complete the command with CHK set to one. The device shall report a Sense key of 04h (HARDWARE ERROR). The sense data supplied via a subsequent REQUEST SENSE command shall report an ASC/ASCQ value of 08h/03h (LOGICAL UNIT COMMUNICATION CRC ERROR). Host drivers should retry the command that resulted in a HARDWARE ERROR.
10) A host may send extra data words on the last Ulotra DMA burst of a data out command. If a device determines that all data has been transferred for a command, the device shall terminate the burst. A devide may have already received more data words that were required for the command. These extra words are used by both the host and the devive to calculate the CRC, but, on an Ultr DMA data-out burst, the extra words shall be discarded by the device.
11) The CRC generator polynomial is: G(X) = X16 + X12 + X5 + 1. **Table 47** describes the equations for 16-bit parallel generation of the resulting polynomial (based on a word boundary).

NOTE – Since no bit clock is available, the recommended approach for calculating CRC is to use a word clock derived from the bus strobe. The combinational logic is then equivalent to shifting sixteen bits serially through the generator polynomial where DD0 is shifted in first and DD15 is shifted in last.

NOTE – If excessive CRC errors are encountered while operating in an Ultra mode, the host should select a slower Ultra mode. Caution: CRC errors are detected and reported only while operating in an Ultra mode.

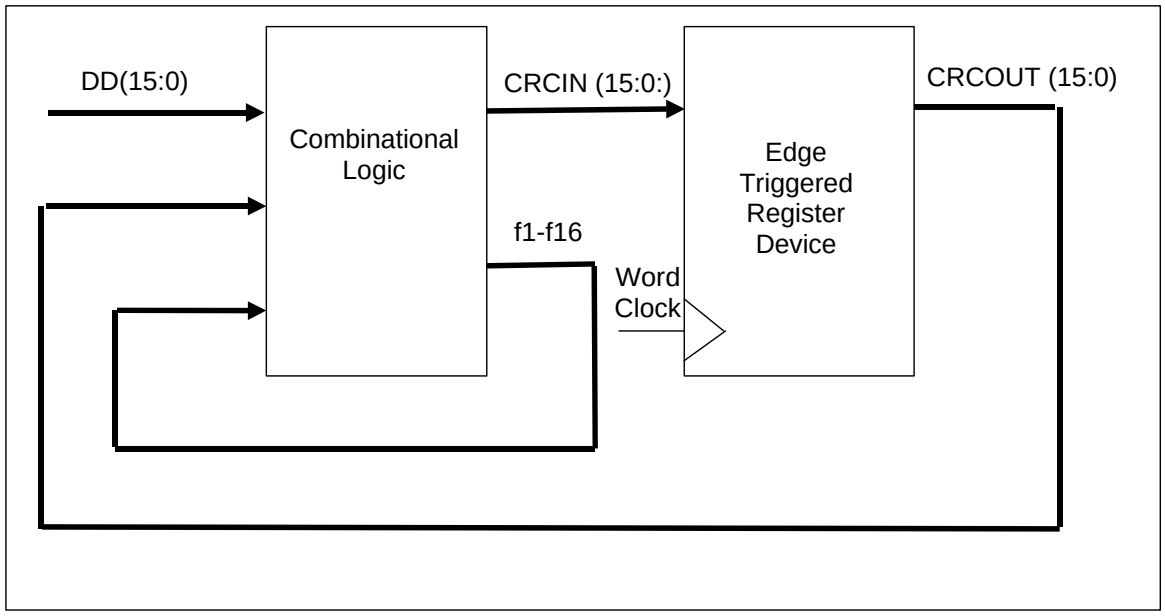



**Figure 64 – Example Parallel CRC generator**

**Table 47 – Equations for parallel generation of a CRC polynomial**

| | |
|---|---|
| CRCIN0 = f16 | CRCIN8 = f8 XOR f13 |
| CRCIN1 = f15 | CRCIN9 = f7 XOR f12 |
| CRCIN2 = f14 | CRCIN10 = f6 XOR f11 |
| CRCIN3 = f13 | CRCIN11 = f5 XOR f10 |
| CRCIN4 = f12 | CRCIN12 = f4 XOR f9 XOR f16 |
| CRCIN5 = f11 XOR f16 | CRCIN13 = f3 XOR f8 XOR f15 |
| CRCIN6 = f10 XOR f15 | CRCIN14 = f2 XOR f7 XOR f14 |
| CRCIN7 = f9 XOR f14 | CRCIN15 = f1 XOR f6 XOR f13 |
| f1 = DD0 XOR CRCOUT15 | f9 = DD8 XOR CRCOUT7 XOR f5 |
| f2 = DD1 XOR CRCOUT14 | f10 = DD9 XOR CRCOUT6 XOR f6 |
| f3 = DD2 XOR CRCOUT13 | f11 = DD10 XOR CRCOUT5 XOR f7 |
| f4 = DD3 XOR CRCOUT12 | f12 = DD11 XOR CRCOUT4 XOR f1 XOR f8 |
| f5 = DD4 XOR CRCOUT11 XOR f1 | f13 = DD12 XOR CRCOUT3 XOR f2 XOR f9 |
| f6 = DD5 XOR CRCOUT10 XOR f2 | f14 = DD13 XOR CRCOUT2 XOR f3 XOR f10 |
| f7 = DD6 XOR CRCOUT9 XOR f3 | f15 = DD14 XOR CRCOUT1 XOR f4 XOR f11 |
| f8 = DD7 XOR CRCOUT8 XOR f4 | f16 = DD15 XOR CRCOUT0 XOR f5 XOR f12 |

NOTES –
1 f = feedback
2 DD = Data to or from the bus
3 CRCOUT = 16-bit edge triggered result (current CRC)
4 CRCOUT(15:0) are sent on matching order bits of DD(15:0)
5 CRCIN = Output of combinatorial logic (next CRC)

# 12 Parallel interface timing

## 12.1 Deskewing

For PIO and Multiword DMA modes all timing values shall be measured at the connector of the selected device. The host shall account for cable skew.

For Ultra DMA modes unless otherwise specified, timing parameters shall be measured at the connector of the host or device to which the parameter applies.

## 12.2 Transfer timing

The minimum cycle time supported by the device in PIO mode 3, 4 and Multiword DMA mode 1, 2 respectively shall always be greater than or equal to the minimum cycle time defined by the associated mode e.g., a device supporting PIO mode 4 timing shall not report a value less than 120 ns, the minimum cycle time defined for PIO mode 4 timings.

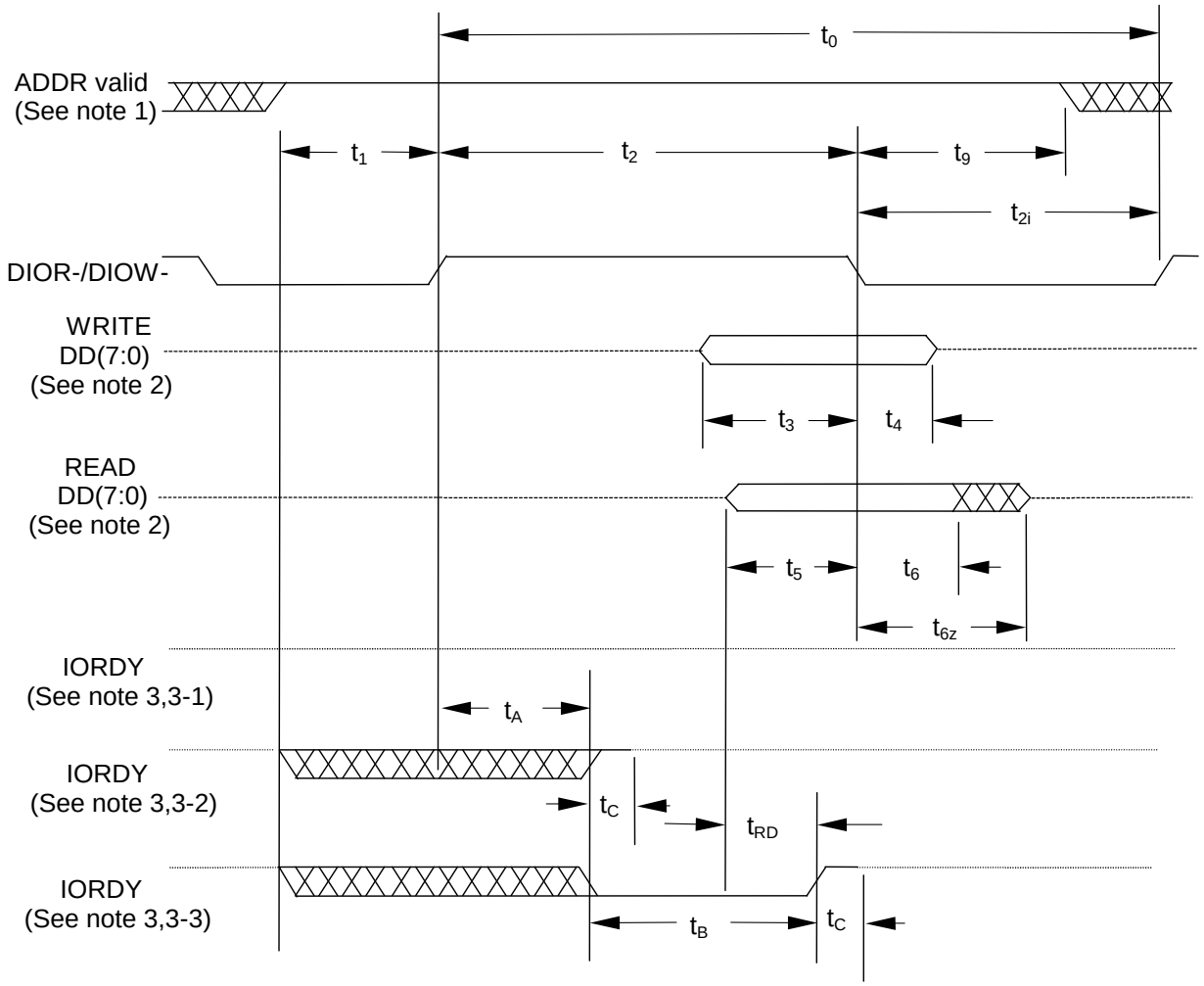
See **Error! Reference source not found.** for timing diagram conventions.

### 12.2.1 Register transfers

Figure 65 defines the relationships between the interface signals for register transfers. Peripherals reporting support for PIO mode 3 or 4 shall power-up in a PIO mode 0, 1, or 2.

For PIO modes 3 and above, the minimum value of $t_0$ is specified by word 68 in the IDENTIFY DEVICE parameter list. Table 48 defines the minimum value that shall be placed in word 68.

Both hosts and devices shall support IORDY when PIO mode 3 or 4 are the currently selected mode of operation.




NOTES –

1 Device address consists of signals CS0-, CS1-, and DA(2:0)

2 Data consists of DD(7:0).

3 The negation of IORDY by the device is used to extend the register transfer cycle. The determination of whether the cycle is to be extended is made by the host after $t_A$ from the assertion of DIOR- or DIOW-. The assertion and negation of IORDY are described in the following three cases:

- 3-1 Device never negates IORDY, devices keeps IORDY released: no wait is generated.
- 3-2 Device negates IORDY before $t_A$, but causes IORDY to be asserted before $t_A$. IORDY is released prior to negation and may be asserted for no more than 5 ns before release: no wait generated.
- 3-3 Device negates IORDY before $t_A$. IORDY is released prior to negation and may be asserted for no more than 5 ns before release: wait generated. The cycle completes after IORDY is reasserted. For cycles where a wait is generated and DIOR- is asserted, the device shall place read data on DD(7:0) for $t_{RD}$ before asserting IORDY.

4 DMACK- shall remain negated during a register transfer.

**Figure 65 – Register transfer to/from device**

**Table 48 – Register transfer to/from device**

| Register transfer timing parameters | | | Mode 0 ns | Mode 1 ns | Mode 2 ns | Mode 3 ns | Mode 4 ns | Note |
|---|---|---|---|---|---|---|---|---|
| $t_0$ | Cycle time | (min) | 600 | 383 | 330 | 180 | 120 | 1,4,5 |
| $t_1$ | Address valid to DIOR-/DIOW- setup | (min) | 70 | 50 | 30 | 30 | 25 | |
| $t_2$ | DIOR-/DIOW- pulse width 8-bit | (min) | 290 | 290 | 290 | 80 | 70 | 1 |
| $t_{2i}$ | DIOR-/DIOW- recovery time | (min) | - | - | - | 70 | 25 | 1 |
| $t_3$ | DIOW- data setup | (min) | 60 | 45 | 30 | 30 | 20 | |
| $t_4$ | DIOW- data hold | (min) | 30 | 20 | 15 | 10 | 10 | |
| $t_5$ | DIOR- data setup | (min) | 50 | 35 | 20 | 20 | 20 | |
| $t_6$ | DIOR- data hold | (min) | 5 | 5 | 5 | 5 | 5 | |
| $t_{6Z}$ | DIOR- data tristate | (max) | 30 | 30 | 30 | 30 | 30 | 2 |
| $t_9$ | DIOR-/DIOW- to address valid hold | (min) | 20 | 15 | 10 | 10 | 10 | |
| $t_{RD}$ | Read Data Valid to IORDY active (if IORDY initially low after $t_A$) | (min) | 0 | 0 | 0 | 0 | 0 | |
| $t_A$ | IORDY Setup time | | 35 | 35 | 35 | 35 | 35 | 3 |
| $t_B$ | IORDY Pulse Width | (max) | 1250 | 1250 | 1250 | 1250 | 1250 | |
| $t_C$ | IORDY assertion to release | (max) | 5 | 5 | 5 | 5 | 5 | |

NOTES –

1 $t_0$ is the minimum total cycle time, $t_2$ is the minimum DIOR-/DIOW- assertion time, and $t_{2i}$ is the minimum DIOR-/DIOW- negation time. A host implementation shall lengthen $t_2$ and/or $t_{2i}$ to ensure that $t_0$ is equal to or greater than the value reported in the devices IDENTIFY DEVICE data. A device implementation shall support any legal host implementation.

2 This parameter specifies the time from the negation edge of DIOR- to the time that the data bus is released by the device.

3 The delay from the activation of DIOR- or DIOW- until the state of IORDY is first sampled. If IORDY is inactive then the host shall wait until IORDY is active before the register transfer cycle is completed. If the device is not driving IORDY negated at the $t_A$ after the activation of DIOR- or DIOW-, then $t_5$ shall be met and $t_{RD}$ is not applicable. If the device is driving IORDY negated at the time $t_A$ after the activation of DIOR- or DIOW-, then $t_{RD}$ shall be met and $t_5$ is not applicable.

4 ATA/ATAPI standards prior to ATA/ATAPI-5 inadvertently specified an incorrect value for mode 2 time $t_0$ by utilizing the 16-bit PIO value

5 Mode shall be selected no higher than the highest mode supported by the slowest device.

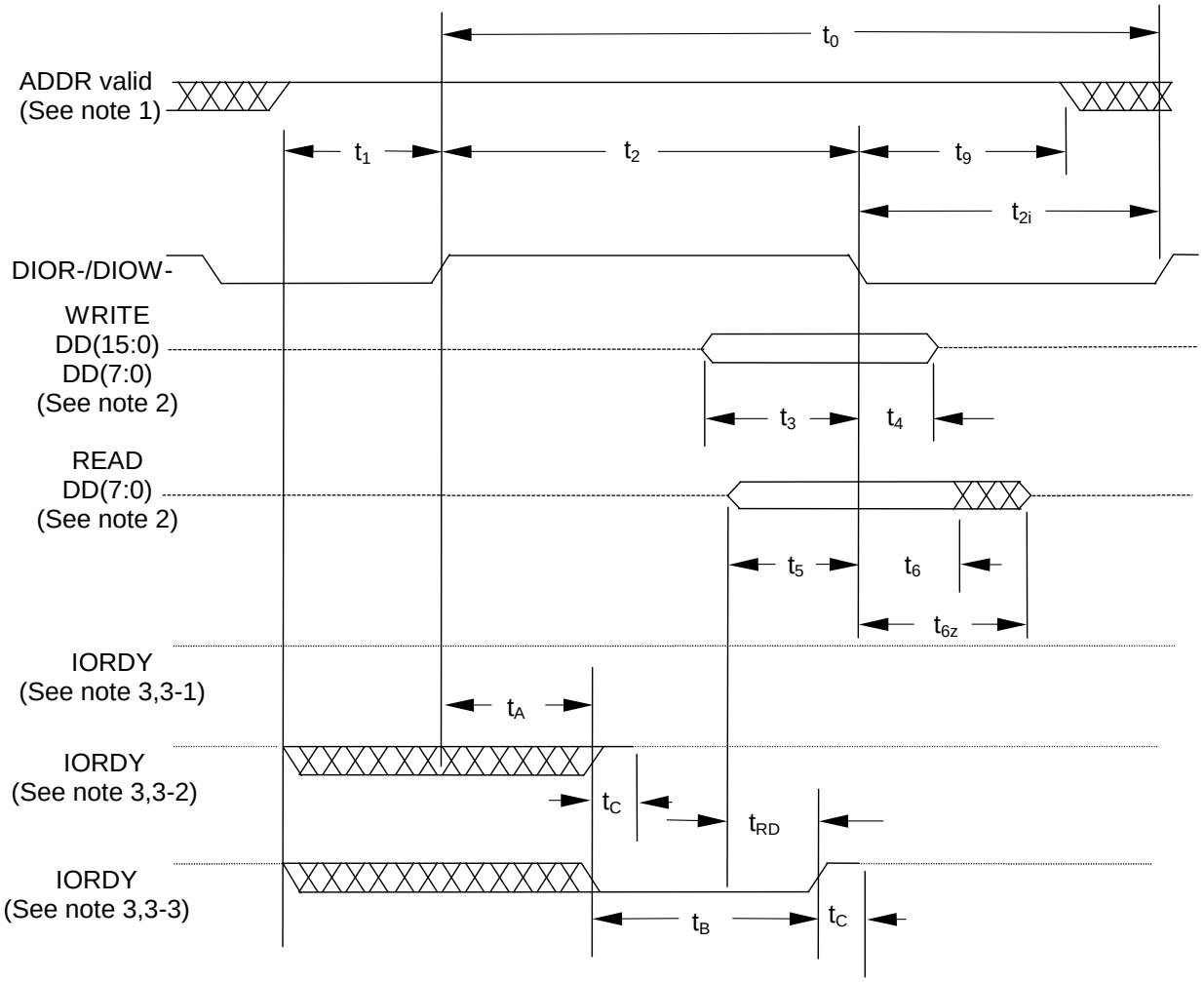
### 12.2.2 PIO data transfers

Figure 66 defines the relationships between the interface signals for PIO data transfers. Peripherals reporting support for PIO mode 3 or 4 shall power-up in a PIO mode 0, 1, or 2.

For PIO modes 3 and above, the minimum value of $t_0$ is specified by word 68 in the IDENTIFY DEVICE parameter list. Table 49 defines the minimum value that shall be placed in word 68.

IORDY shall be supported when PIO mode 3 or 4 are the current mode of operation.

NOTE – Some devices implementing the PACKET Command feature set prior to ATA/ATAPI-4 power-up in PIO mode 3 and enable IORDY as the default.




NOTES –

1 Device address consists of signals CS0-, CS1-, and DA(2:0)

2 Data consists of DD(7:0).

3 The negation of IORDY by the device is used to extend the register transfer cycle. The determination of whether the cycle is to be extended is made by the host after $t_A$ from the assertion of DIOR- or DIOW-. The assertion and negation of IORDY are described in the following three cases:

- 3-1 Device never negates IORDY, devices keeps IORDY released: no wait is generated.
- 3-2 Device negates IORDY before $t_A$, but causes IORDY to be asserted before $t_A$. IORDY is released prior to negation and may be asserted for no more than 5 ns before release: no wait generated.
- 3-3 Device negates IORDY before $t_A$. IORDY is released prior to negation and may be asserted for no more than 5 ns before release: wait generated. The cycle completes after IORDY is reasserted. For cycles where a wait is generated and DIOR- is asserted, the device shall place read data on DD(7:0) for $t_{RD}$ before asserting IORDY.

4 DMACK- shall remain negated during a register transfer.

**Figure 66 – PIO data transfer to/from device**

**Table 49 – PIO data transfer to/from device**

| PIO timing parameters | | | Mode 0 ns | Mode 1 ns | Mode 2 ns | Mode 3 ns | Mode 4 ns | Note |
|---|---|---|---|---|---|---|---|---|
| $t_0$ | Cycle time | (min) | 600 | 383 | 240 | 180 | 120 | 1,4 |
| $t_1$ | Address valid to DIOR-/DIOW- setup | (min) | 70 | 50 | 30 | 30 | 25 | |
| $t_2$ | DIOR-/DIOW- | (min) | 165 | 125 | 100 | 80 | 70 | 1 |
| $t_{2i}$ | DIOR-/DIOW- recovery time | (min) | - | - | - | 70 | 25 | 1 |
| $t_3$ | DIOW- data setup | (min) | 60 | 45 | 30 | 30 | 20 | |
| $t_4$ | DIOW- data hold | (min) | 30 | 20 | 15 | 10 | 10 | |
| $t_5$ | DIOR- data setup | (min) | 50 | 35 | 20 | 20 | 20 | |
| $t_6$ | DIOR- data hold | (min) | 5 | 5 | 5 | 5 | 5 | |
| $t_{6Z}$ | DIOR- data tristate | (max) | 30 | 30 | 30 | 30 | 30 | 2 |
| $t_9$ | DIOR-/DIOW- to address valid hold | (min) | 20 | 15 | 10 | 10 | 10 | |
| $t_{RD}$ | Read Data Valid to IORDY active (if IORDY initially low after $t_A$) | (min) | 0 | 0 | 0 | 0 | 0 | |
| $t_A$ | IORDY Setup time | | 35 | 35 | 35 | 35 | 35 | 3 |
| $t_B$ | IORDY Pulse Width | (max) | 1250 | 1250 | 1250 | 1250 | 1250 | |
| $t_C$ | IORDY assertion to release | (max) | 5 | 5 | 5 | 5 | 5 | |

NOTES –

1 $t_0$ is the minimum total cycle time, $t_2$ is the minimum DIOR-/DIOW- assertion time, and $t_{2i}$ is the minimum DIOR-/DIOW- negation time. A host implementation shall lengthen $t_2$ and/or $t_{2i}$ to ensure that $t_0$ is equal to or greater than the value reported in the devices IDENTIFY DEVICE data. A device implementation shall support any legal host implementation.

2 This parameter specifies the time from the negation edge of DIOR- to the time that the data bus is released by the device.

3 The delay from the activation of DIOR- or DIOW- until the state of IORDY is first sampled. If IORDY is inactive then the host shall wait until IORDY is active before the PIO cycle is completed. If the device is not driving IORDY negated at the $t_A$ after the activation of DIOR- or DIOW-, then $t_5$ shall be met and $t_{RD}$ is not applicable. If the device is driving IORDY negated at the time $t_A$ after the activation of DIOR- or DIOW-, then $t_{RD}$ shall be met and $t_5$ is not applicable.

4 Mode may be selected at the highest mode for the device if CS(1:0) and DA(2:0) do not change between read or write cycles or selected at the highest mode supported by the slowest device if CS(1:0) or DA(2:0) do change between read or write cycles.

### 12.2.3 Multiword DMA data transfer

Figure 67 through Figure 70 define the timing associated with Multiword DMA transfers.

For Multiword DMA modes 1 and above, the minimum value of $t_0$ is specified by word 65 in the IDENTIFY DEVICE parameter list. Table 50 defines the minimum value that shall be placed in word 65.

Devices shall power-up with mode 0 as the default Multiword DMA mode.

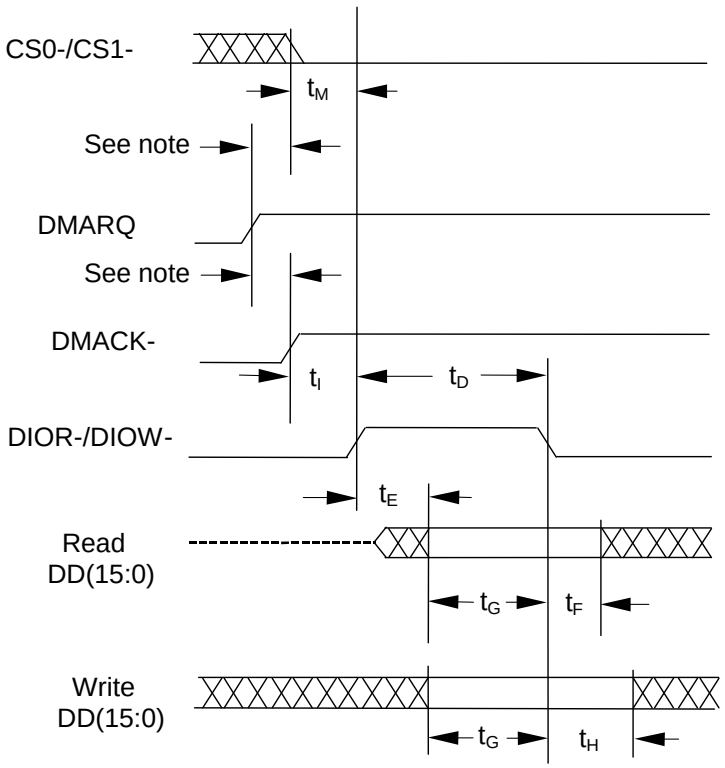
**Table 50 – Multiword DMA data transfer**

| Multiword DMA timing parameters | | | Mode 0 ns | Mode 1 ns | Mode 2 ns | Note |
|---|---|---|---|---|---|---|
| $t_0$ | Cycle time | (min) | 480 | 150 | 120 | see note |
| $t_D$ | DIOR-/DIOW- asserted pulse width | (min) | 215 | 80 | 70 | see note |
| $t_E$ | DIOR- data access | (max) | 150 | 60 | 50 | |
| $t_F$ | DIOR- data hold | (min) | 5 | 5 | 5 | |
| $t_G$ | DIOR-/DIOW- data setup | (min) | 100 | 30 | 20 | |
| $t_H$ | DIOW- data hold | (min) | 20 | 15 | 10 | |
| $t_I$ | DMACK to DIOR-/DIOW- setup | (min) | 0 | 0 | 0 | |
| $t_J$ | DIOR-/DIOW- to DMACK hold | (min) | 20 | 5 | 5 | |
| $t_{KR}$ | DIOR- negated pulse width | (min) | 50 | 50 | 25 | see note |
| $t_{KW}$ | DIOW- negated pulse width | (min) | 215 | 50 | 25 | see note |
| $t_{LR}$ | DIOR- to DMARQ delay | (max) | 120 | 40 | 35 | |
| $t_{LW}$ | DIOW- to DMARQ delay | (max) | 40 | 40 | 35 | |
| $t_M$ | CS(1:0) valid to DIOR-/DIOW- | (min) | 50 | 30 | 25 | |
| $t_N$ | CS(1:0) hold | (min) | 15 | 10 | 10 | |
| $t_Z$ | DMACK- to read data released | (max) | 20 | 25 | 25 | |

NOTE – $t_0$ is the minimum total cycle time, $t_D$ is the minimum DIOR-/DIOW- assertion time, and $t_K$ ($t_{KR}$ or $t_{KW}$, as appropriate) is the minimum DIOR-/DIOW- negation time. A host shall lengthen $t_D$ and/or $t_K$ to ensure that $t_0$ is equal to the value reported in the devices IDENTIFY DEVICE data.

#### 12.2.3.1 Initiating a Multiword DMA data burst

The values for the timings for each of the Multiword DMA modes are contained in Table 50.




NOTE – The host shall not assert DMACK- or negate both CS0 and CS1 until the assertion of DMARQ is detected. The maximum time from the assertion of DMARQ to the assertion of DMACK- or the negation of both CS0 and CS1 is not defined.

**Figure 67 – Initiating a Multiword DMA data burst**

#### 12.2.3.2 Sustaining a Multiword DMA data burst

The values for the timings for each of the Multiword DMA modes are contained in Table 50.

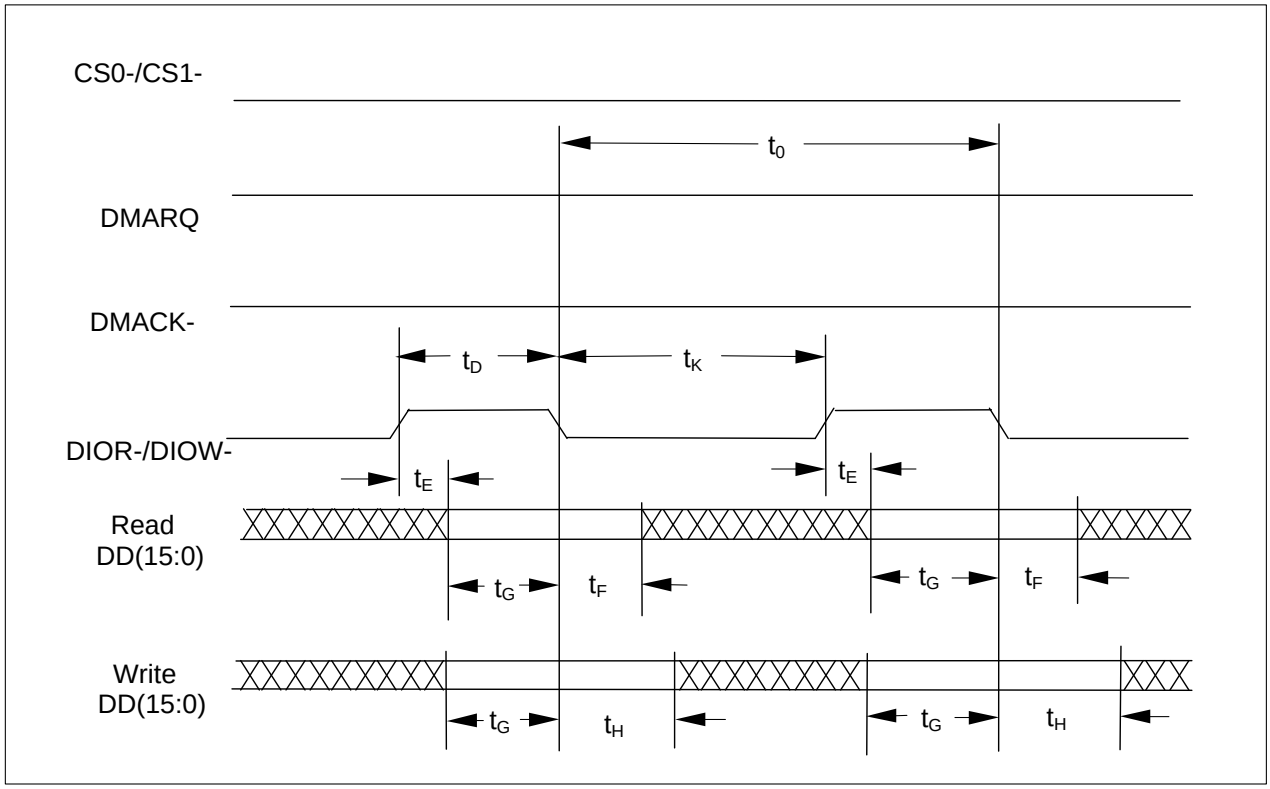
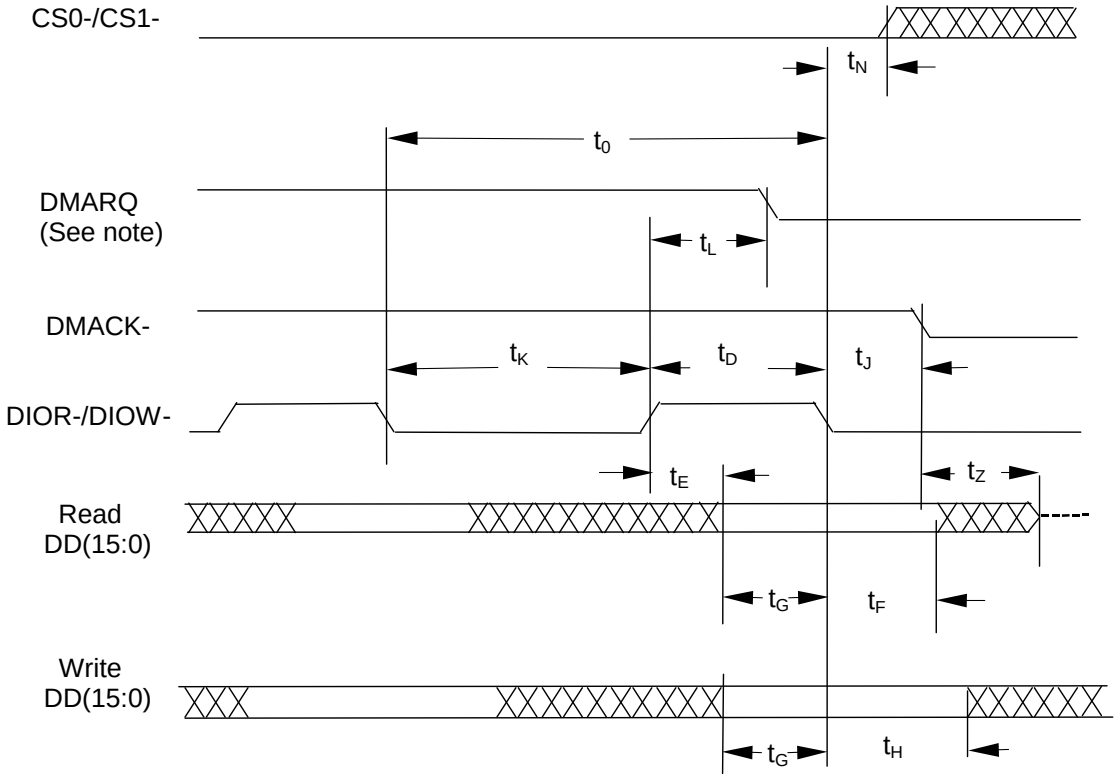



**Figure 68 - Sustaining a Multiword DMA data burst**

#### 12.2.3.3 Device terminating a Multiword DMA data burst

The values for the timings for each of the Multiword DMA modes are contained in Table 50.

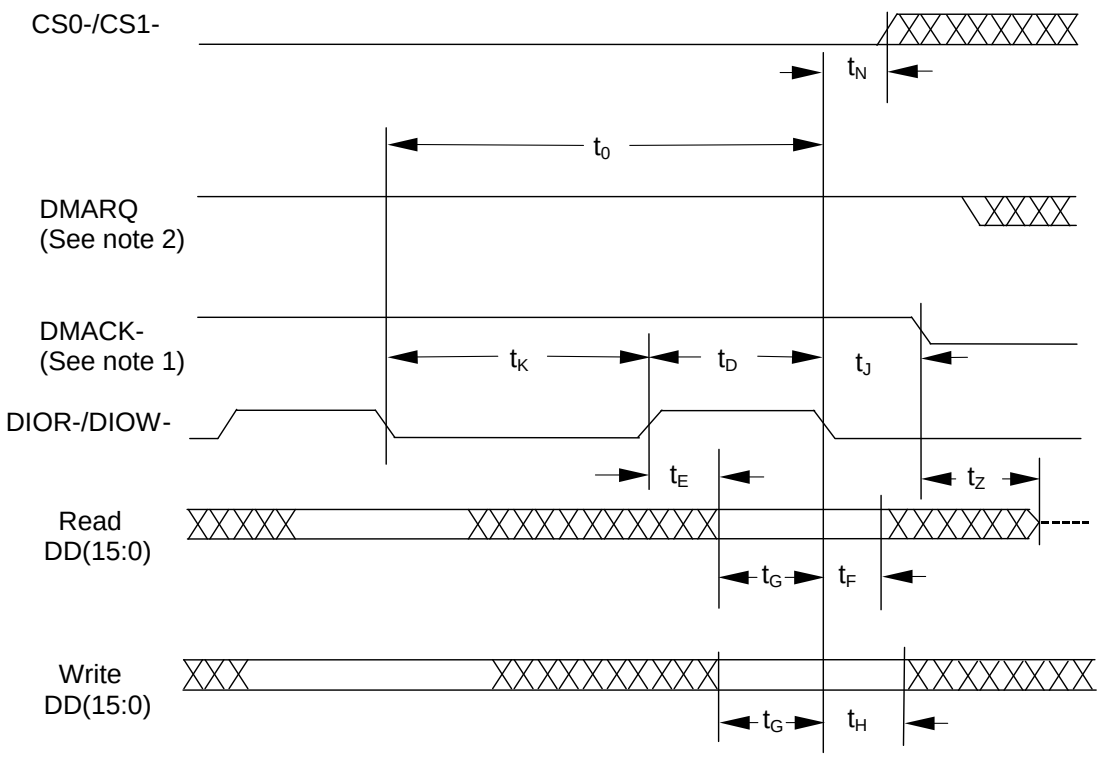



NOTE – To terminate the data burst, the Device shall negate DMARQ within $t_L$ of the assertion of the current DIOR- or DIOW- pulse. The last data word for the burst shall then be transferred by the negation of the current DIOR- or DIOW- pulse. If all data for the command has not been transferred, the device shall reassert DMARQ again at any later time to resume the DMA operation as shown in figure 66.

**Figure 69 – Device terminating a Multiword DMA data burst**

#### 12.2.3.4 Host terminating a Multiword DMA data burst

The values for the timings for each of the Multiword DMA modes are contained in Table 50.




NOTE –
1 To terminate the transmission of a data burst, the host shall negate DMACK- within $t_J$ after a DIOR- or DIOW- pulse. No further DIOR- or DIOW- pulses shall be asserted for this burst.
2 If the device is able to continue the transfer of data, the device may leave DMARQ asserted and wait for the host to reassert DMACK- or may negate DMARQ at any time after detecting that DMACK- has been negated.

**Figure 70 – Host terminating a Multiword DMA data burst**

### 12.2.4 Ultra DMA data transfer

Figure 71 through Figure 80 define the timings associated with all phases of Ultra DMA bursts.

Table **51** contains the values for the timings for each of the Ultra DMA modes. Table 52 contains descriptions and comments for each of the timing values in
Table **51**. Table 53 contains timings specified for the IC alone.

All timings are worst case across functional voltage, process, temperature, and system configuration variances.

**Table 51 – Ultra DMA data burst timing requirements**

| Name | Mode 0 (in ns) | | Mode 1 (in ns) | | Mode 2 (in ns) | | Mode 3 (in ns) | | Mode 4 (in ns) | | Mode 5 (in ns) | | Mode 6 (in ns) | | Measurement location |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Min | Max | Min | Max | Min | Max | Min | Max | Min | Max | Min | Max | Min | Max | |
| $t_{2CYCTYP}$ | 240 | | 160 | | 120 | | 90 | | 60 | | 40 | | 30 | | Sender |
| $t_{CYC}$ | 112 | | 73 | | 54 | | 39 | | 25 | | 16.8 | | 13.0 | | Note 3 |
| $t_{2CYC}$ | 230 | | 153 | | 115 | | 86 | | 57 | | 38 | | 29 | | Sender |
| $t_{DS}$ | 15.0 | | 10.0 | | 7.0 | | 7.0 | | 5.0 | | 4.0 | | 2.6 | | Recipient |
| $t_{DH}$ | 5.0 | | 5.0 | | 5.0 | | 5.0 | | 5.0 | | 4.6 | | 3.5 | | Recipient |
| $t_{DVS}$ | 70.0 | | 48.0 | | 31.0 | | 20.0 | | 6.7 | | 4.8 | | 4.0 | | Sender |
| $t_{DVH}$ | 6.2 | | 6.2 | | 6.2 | | 6.2 | | 6.2 | | 4.8 | | 4.0 | | Sender |
| $t_{CS}$ | 15.0 | | 10.0 | | 7.0 | | 7.0 | | 5.0 | | 5.0 | | 5.0 | | Device |
| $t_{CH}$ | 5.0 | | 5.0 | | 5.0 | | 5.0 | | 5.0 | | 5.0 | | 5.0 | | Device |
| $t_{CVS}$ | 70.0 | | 48.0 | | 31.0 | | 20.0 | | 6.7 | | 10.0 | | 10.0 | | Host |
| $t_{CVH}$ | 6.2 | | 6.2 | | 6.2 | | 6.2 | | 6.2 | | 10.0 | | 10.0 | | Host |
| $t_{ZFS}$ | 0 | | 0 | | 0 | | 0 | | 0 | | 35 | | 25 | | Device |
| $t_{DZFS}$ | 70.0 | | 48.0 | | 31.0 | | 20.0 | | 6.7 | | 25 | | 17.5 | | Sender |
| $t_{FS}$ | | 230 | | 200 | | 170 | | 130 | | 120 | | 90 | | 80 | Device |
| $t_{LI}$ | 0 | 150 | 0 | 150 | 0 | 150 | 0 | 100 | 0 | 100 | 0 | 75 | 0 | 60 | Note 4 |
| $t_{MLI}$ | 20 | | 20 | | 20 | | 20 | | 20 | | 20 | | 20 | | Host |
| $t_{UI}$ | 0 | | 0 | | 0 | | 0 | | 0 | | 0 | | 0 | | Host |
| $t_{AZ}$ | | 10 | | 10 | | 10 | | 10 | | 10 | | 10 | | 10 | Note 5 |
| $t_{ZAH}$ | 20 | | 20 | | 20 | | 20 | | 20 | | 20 | | 20 | | Host |
| $t_{ZAD}$ | 0 | | 0 | | 0 | | 0 | | 0 | | 0 | | 0 | | Device |
| $t_{ENV}$ | 20 | 70 | 20 | 70 | 20 | 70 | 20 | 55 | 20 | 55 | 20 | 50 | 20 | 50 | Host |
| $t_{RFS}$ | | 75 | | 70 | | 60 | | 60 | | 60 | | 50 | | 50 | Sender |
| $t_{RP}$ | 160 | | 125 | | 100 | | 100 | | 100 | | 85 | | 85 | | Recipient |
| $t_{IORDYZ}$ | | 20 | | 20 | | 20 | | 20 | | 20 | | 20 | | 20 | Device |
| $t_{ZIORDY}$ | 0 | | 0 | | 0 | | 0 | | 0 | | 0 | | 0 | | Device |
| $t_{ACK}$ | 20 | | 20 | | 20 | | 20 | | 20 | | 20 | | 20 | | Host |
| $t_{SS}$ | 50 | | 50 | | 50 | | 50 | | 50 | | 50 | | 50 | | Sender |

NOTES –
1 All timing measurement switching points (low to high and high to low) shall be taken at 1.5 V.
2 All signal transitions for a timing parameter shall be measured at the connector specified in the measurement location column. For example, in the case of $t_{RFS}$, both STROBE and DMARDY- transitions are measured at the sender connector.
3 The parameter $t_{CYC}$ shall be measured at the recipient's connector farthest from the sender.
4 The parameter $t_{LI}$ shall be measured at the connector of the sender or recipient that is responding to an incoming transition from the recipient or sender respectively. Both the incoming signal and the outgoing response shall be measured at the same connector.
5 The parameter $t_{AZ}$ shall be measured at the connector of the sender or recipient that is driving the bus but must release the bus the allow for a bus turnaround.

**Table 52 – Ultra DMA data burst timing descriptions**

| Name | Comment |
|---|---|
| $t_{2CYCTYP}$ | Typical sustained average two cycle time |
| $t_{CYC}$ | Cycle time allowing for asymmetry and clock variations (from STROBE edge to STROBE edge) |
| $t_{2CYC}$ | Two cycle time allowing for clock variations (from rising edge to next rising edge or from falling edge to next falling edge of STROBE) |
| $t_{DS}$ | Data setup time at recipient (from data valid until STROBE edge) (see note 2,5) |
| $t_{DH}$ | Data hold time at recipient (from STROBE edge until data may become invalid) (see note 2,5) |
| $t_{DVS}$ | Data valid setup time at sender (from data valid until STROBE edge) (see note 3) |
| $t_{DVH}$ | Data valid hold time at sender (from STROBE edge until data may become invalid) (see note 3) |
| $t_{CS}$ | CRC word setup time at device (see note 2) |
| $t_{CH}$ | CRC word hold time device (see note 2) |
| $t_{CVS}$ | CRC word valid setup time at host (from CRC valid until DMACK- negation) (see note 3) |
| $t_{CVH}$ | CRC word valid hold time at sender (from DMACK- negation until CRC may become invalid) (see note 3) |
| $t_{ZFS}$ | Time from STROBE output released-to-driving until the first transition of critical timing. |
| $t_{DZFS}$ | Time from data output released-to-driving until the first transition of critical timing. |
| $t_{FS}$ | First STROBE time (for device to first negate DSTROBE from STOP during a data in burst) |
| $t_{LI}$ | Limited interlock time (see note 1) |
| $t_{MLI}$ | Interlock time with minimum (see note 1) |
| $t_{UI}$ | Unlimited interlock time (see note 1) |
| $t_{AZ}$ | Maximum time allowed for output drivers to release (from asserted or negated) |
| $t_{ZAH}$ | Minimum delay time required for output |
| $t_{ZAD}$ | drivers to assert or negate (from released) |
| $t_{ENV}$ | Envelope time (from DMACK- to STOP and HDMARDY- during data in burst initiation and from DMACK to STOP during data out burst initiation) |
| $t_{RFS}$ | Ready-to-final-STROBE time (no STROBE edges shall be sent this long after negation of DMARDY-) |
| $t_{RP}$ | Ready-to-pause time (that recipient shall wait to pause after negating DMARDY-) |
| $t_{IORDYZ}$ | Maximum time before releasing IORDY |
| $t_{ZIORDY}$ | Minimum time before driving IORDY (see note 4) |
| $t_{ACK}$ | Setup and hold times for DMACK- (before assertion or negation) |
| $t_{SS}$ | Time from STROBE edge to negation of DMARQ or assertion of STOP (when sender terminates a burst) |

NOTES –

1 The parameters $t_{UI}$, $t_{MLI}$ (in Figure 74 and Figure 75), and $t_{LI}$ indicate sender-to-recipient or recipient-to-sender interlocks, i.e., one agent (either sender or recipient) is waiting for the other agent to respond with a signal before proceeding. $t_{UI}$ is an unlimited interlock that has no maximum time value. $t_{MLI}$ is a limited time-out that has a defined minimum. $t_{LI}$ is a limited time-out that has a defined maximum.

2 80-conductor cabling (see 7.3) shall be required in order to meet setup ($t_{DS}$, $t_{CS}$) and hold ($t_{DH}$, $t_{CH}$) times in modes greater than 2.

3 Timing for $t_{DVS}$, $t_{DVH}$, $t_{CVS}$ and $t_{CVH}$ shall be met for lumped capacitive loads of 15 and 40 pf at the connector where the Data and STROBE signals have the same capacitive load value. Due to reflections on the cable, these timing measurements are not valid in a normally functioning system.

4 For all modes the parameter $t_{ZIORDY}$ may be greater than $t_{ENV}$ due to the fact that the host has a pull-up on IORDY- giving it a known state when released.

5 The parameters $t_{DS}$, and $t_{DH}$ for mode 5 are defined for a recipient at the end of the cable only in a configuration with a single device located at the end of the cable. This could result in the minimum values for $t_{DS}$ and $t_{DH}$ for mode 5 at the middle connector being 3.0 and 3.9 ns respectively.

**Table 53 – Ultra DMA sender and recipient IC timing requirements**

| Name | Mode 0 (in ns) | | Mode 1 (in ns) | | Mode 2 (in ns) | | Mode 3 (in ns) | | Mode 4 (in ns) | | Mode 5 (in ns) | | Mode 6 (in ns) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Min** | **Max** | **Min** | **Max** | **Min** | **Max** | **Min** | **Max** | **Min** | **Max** | **Min** | **Max** | **Min** | **Max** |
| $t_{DSIC}$ | 14.7 | | 9.7 | | 6.8 | | 6.8 | | 4.8 | | 2.3 | | 2.3 | |
| $t_{DHIC}$ | 4.8 | | 4.8 | | 4.8 | | 4.8 | | 4.8 | | 2.8 | | 2.8 | |
| $t_{DVSIC}$ | 72.9 | | 50.9 | | 33.9 | | 22.6 | | 9.5 | | 6.0 | | 5.2 | |
| $t_{DVHIC}$ | 9.0 | | 9.0 | | 9.0 | | 9.0 | | 9.0 | | 6.0 | | 5.2 | |

| | Comment |
|---|---|
| $t_{DSIC}$ | Recipient IC data setup time (from data valid until STROBE edge) (see note 2) |
| $t_{DHIC}$ | Recipient IC data hold time (from STROBE edge until data may become invalid) (see note 2) |
| $t_{DVSIC}$ | Sender IC data valid setup time (from data valid until STROBE edge) (see note 3) |
| $t_{DVHIC}$ | Sender IC data valid hold time (from STROBE edge until data may become invalid) (see note 3) |

NOTES –

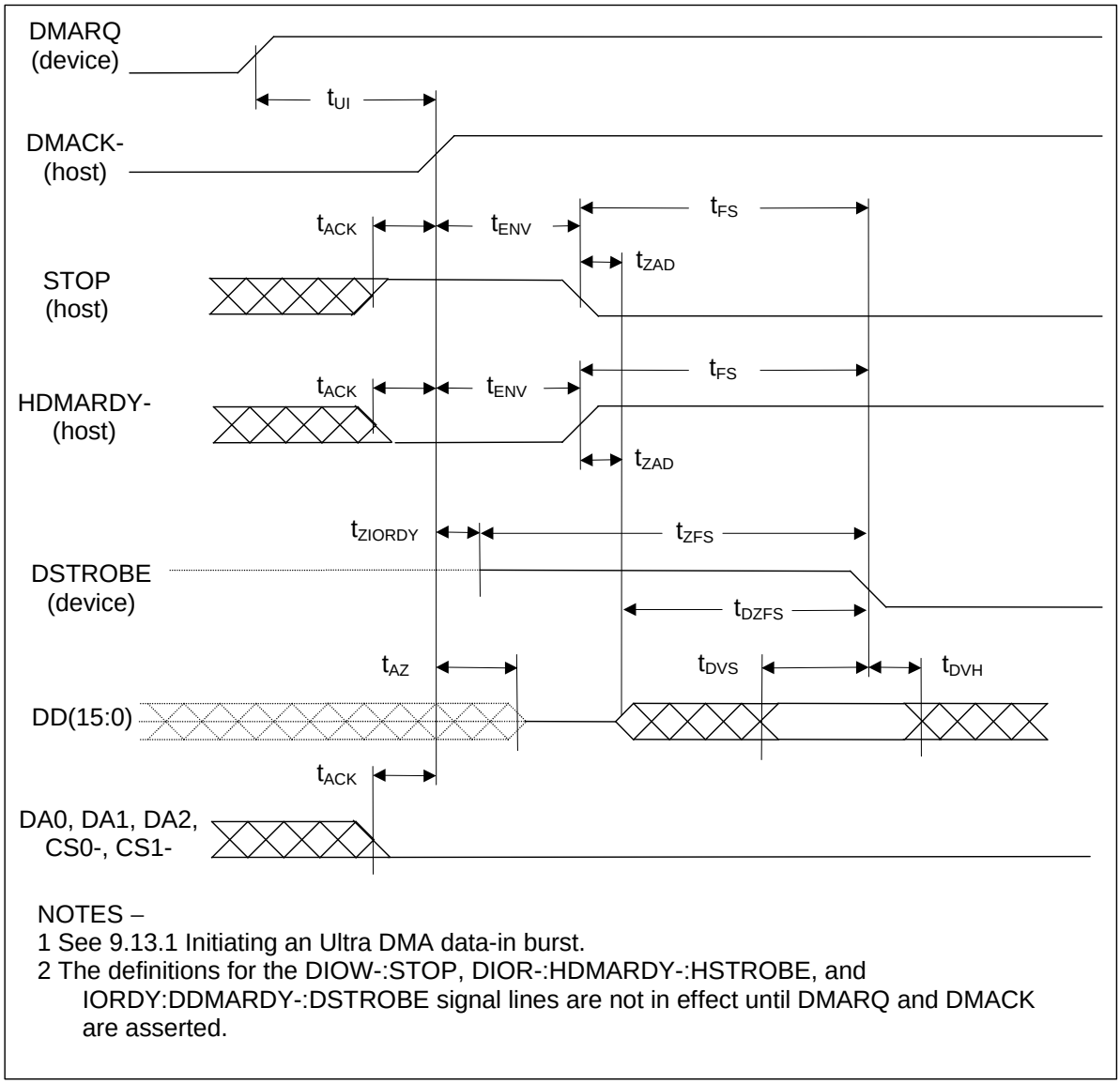
1 All timing measurement switching points(low to high and high to low) shall be taken at 1.5 V.

2 The correct data value shall be captured by the recipient given input data with a slew rate of 0.4 V/ns rising and falling and the input STROBE with a slew rate of 0.4 V/ns rising and falling at $t_{DSIC}$ and $t_{DHIC}$ timing (as measured through 1.5 V).

2 The parameters $t_{DVSIC}$ and $t_{DVHIC}$ shall be met for lumped capacitive loads of 15 and 40 pf at the IC where all signals have the same capacitive load value. Noise that may couple onto the output signals from external sources has not been included in these values.

#### 12.2.4.1 Initiating an Ultra DMA data-in burst

The values for the timings for each of the Ultra DMA modes are contained in 12.2.4.

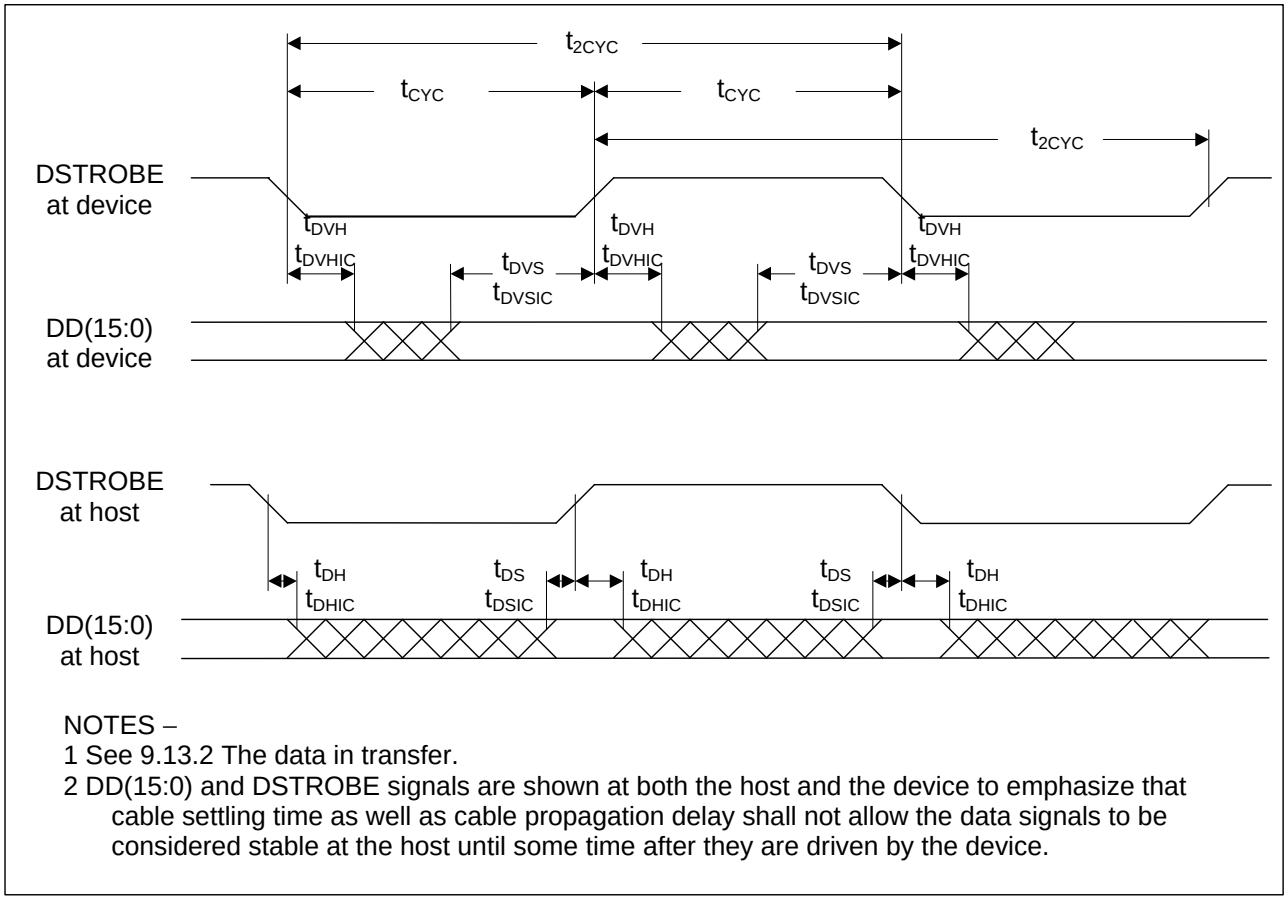



NOTES –
1 See 9.13.1 Initiating an Ultra DMA data-in burst.
2 The definitions for the DIOW-:STOP, DIOR-:HDMARDY-:HSTROBE, and IORDY:DDMARDY-:DSTROBE signal lines are not in effect until DMARQ and DMACK are asserted.

**Figure 71 – Initiating an Ultra DMA data-in burst**

#### 12.2.4.2 Sustained Ultra DMA data-in burst

The values for the timings for each of the Ultra DMA modes are contained in 12.2.4.

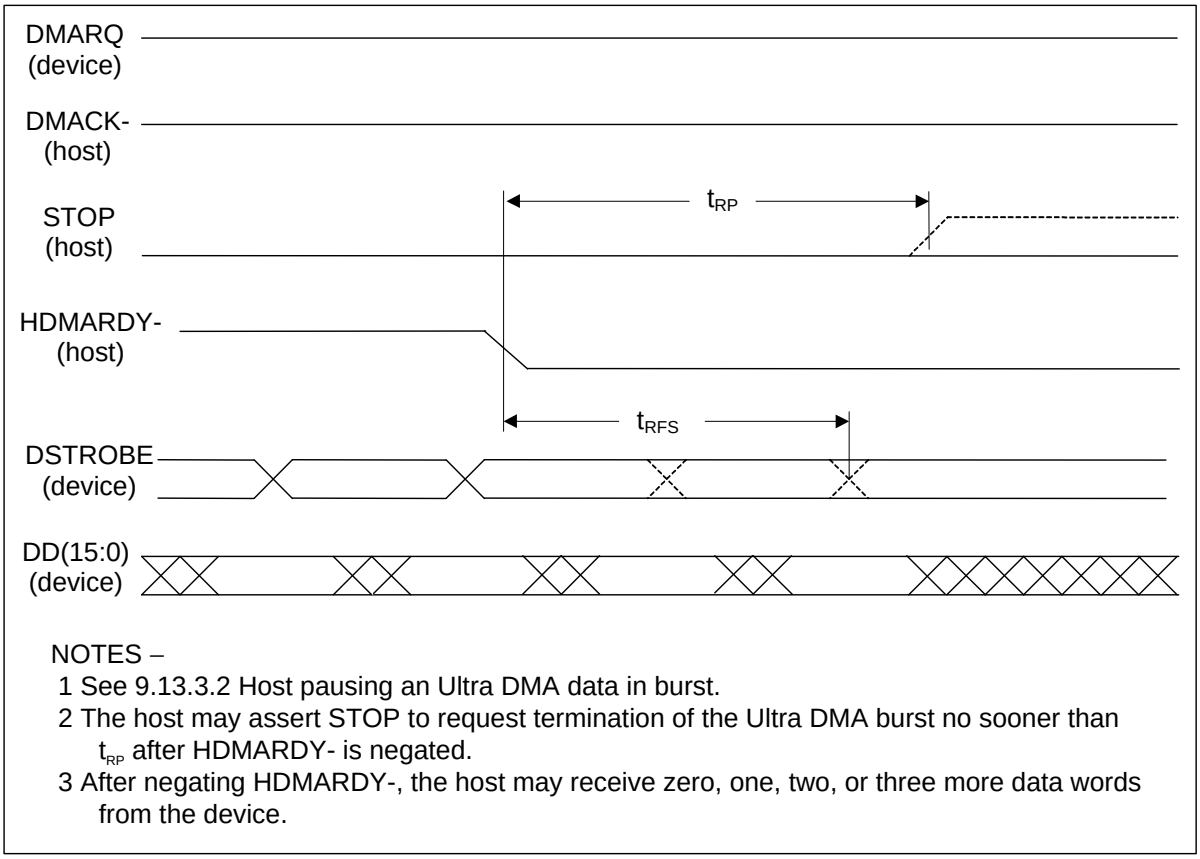



NOTES –
1 See 9.13.2 The data in transfer.
2 DD(15:0) and DSTROBE signals are shown at both the host and the device to emphasize that cable settling time as well as cable propagation delay shall not allow the data signals to be considered stable at the host until some time after they are driven by the device.

**Figure 72 – Sustained Ultra DMA data-in burst**

#### 12.2.4.3 Host pausing an Ultra DMA data-in burst

The values for the timings for each of the Ultra DMA modes are contained in 12.2.4.

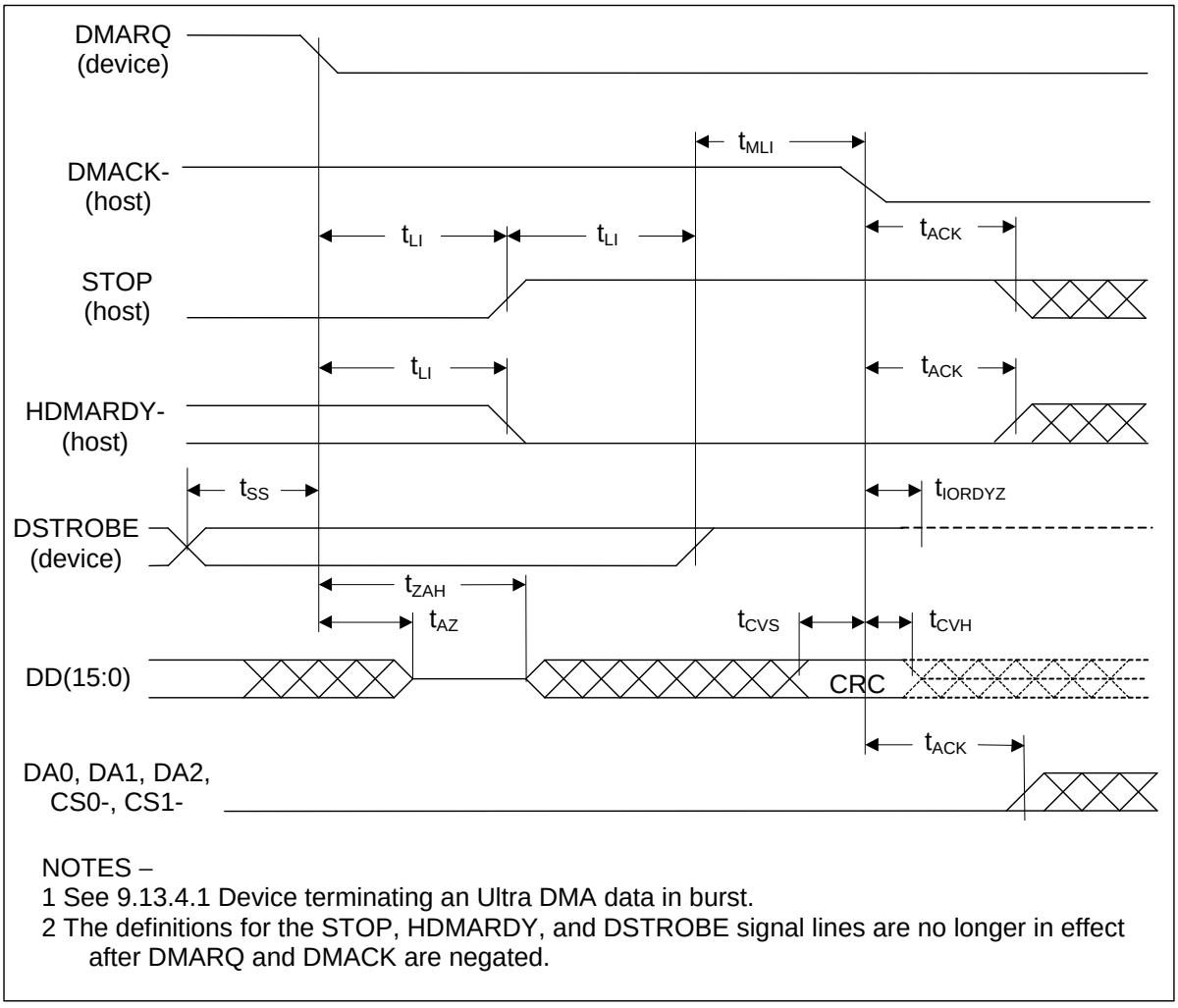



NOTES –
1 See 9.13.3.2 Host pausing an Ultra DMA data in burst.
2 The host may assert STOP to request termination of the Ultra DMA burst no sooner than $t_{RP}$ after HDMARDY- is negated.
3 After negating HDMARDY-, the host may receive zero, one, two, or three more data words from the device.

**Figure 73 – Host pausing an Ultra DMA data-in burst**

#### 12.2.4.4 Device terminating an Ultra DMA data-in burst

The values for the timings for each of the Ultra DMA modes are contained in 12.2.4.

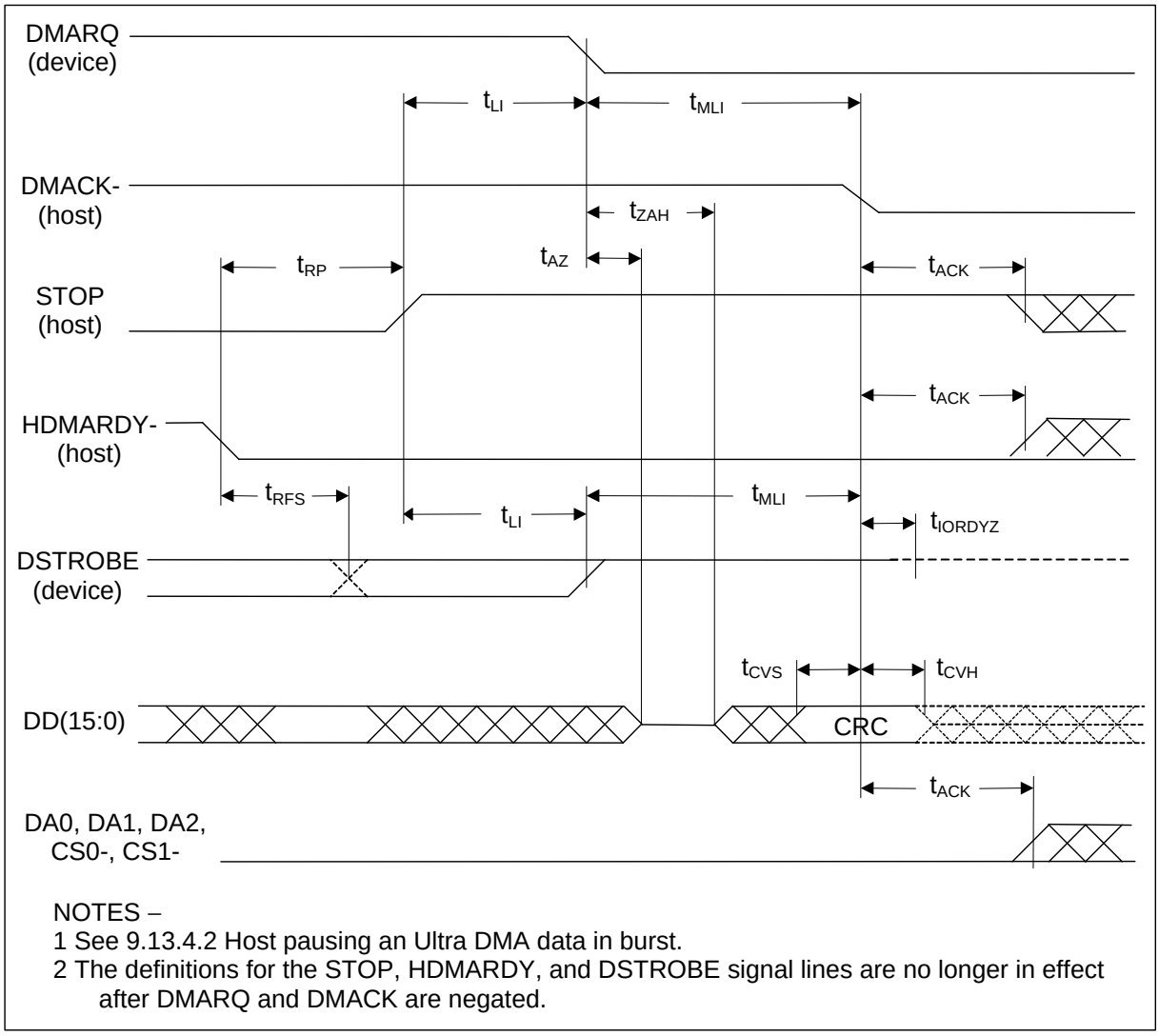



NOTES –
1 See 9.13.4.1 Device terminating an Ultra DMA data in burst.
2 The definitions for the STOP, HDMARDY, and DSTROBE signal lines are no longer in effect after DMARQ and DMACK are negated.

**Figure 74 – Device terminating an Ultra DMA data-in burst**

#### 12.2.4.5 Host terminating an Ultra DMA data-in burst

The values for the timings for each of the Ultra DMA modes are contained in 12.2.4.

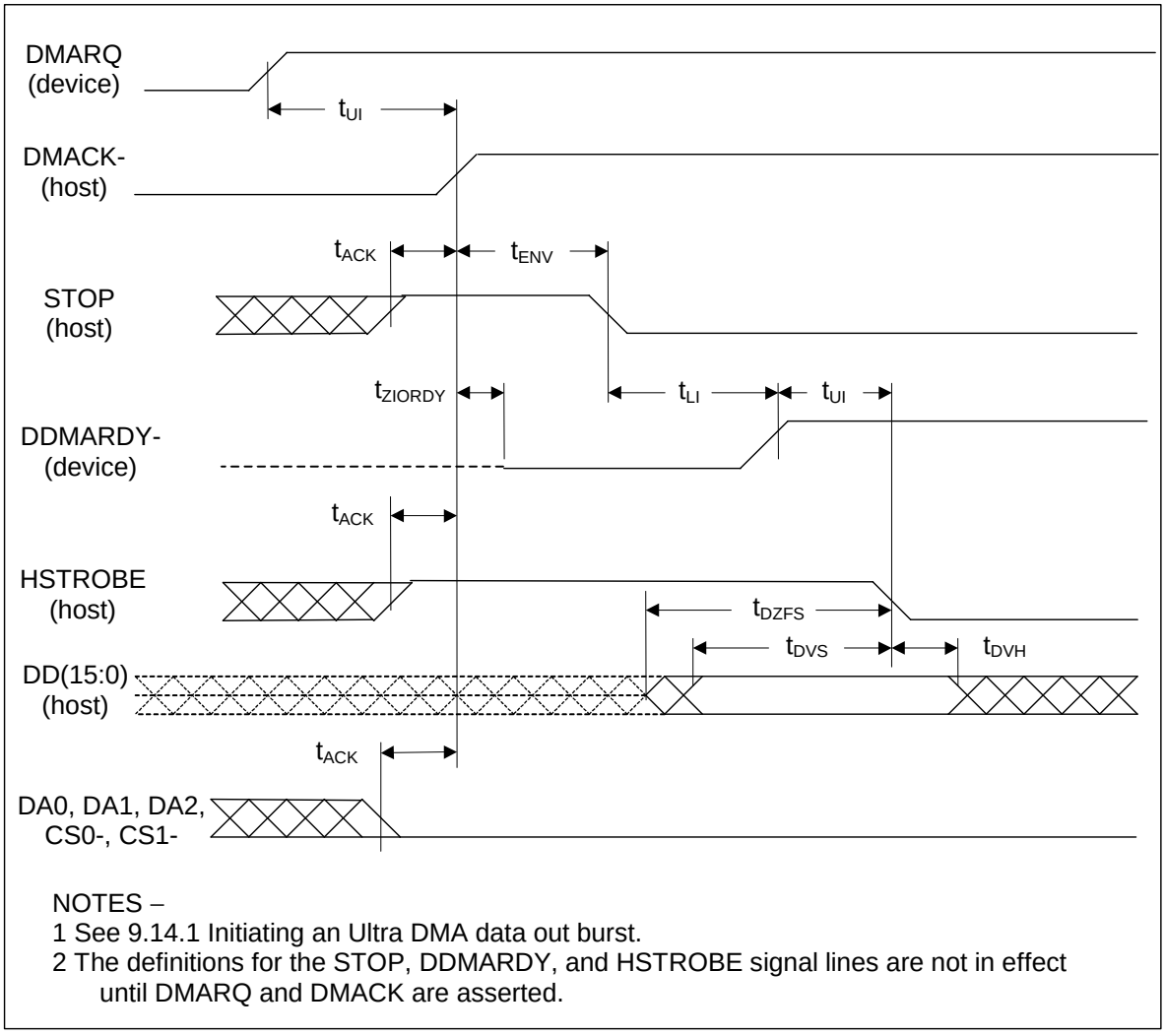



NOTES –
1 See 9.13.4.2 Host pausing an Ultra DMA data in burst.
2 The definitions for the STOP, HDMARDY, and DSTROBE signal lines are no longer in effect after DMARQ and DMACK are negated.

**Figure 75 – Host terminating an Ultra DMA data-in burst**

#### 12.2.4.6 Initiating an Ultra DMA data-out burst

The values for the timings for each of the Ultra DMA modes are contained in 12.2.4.

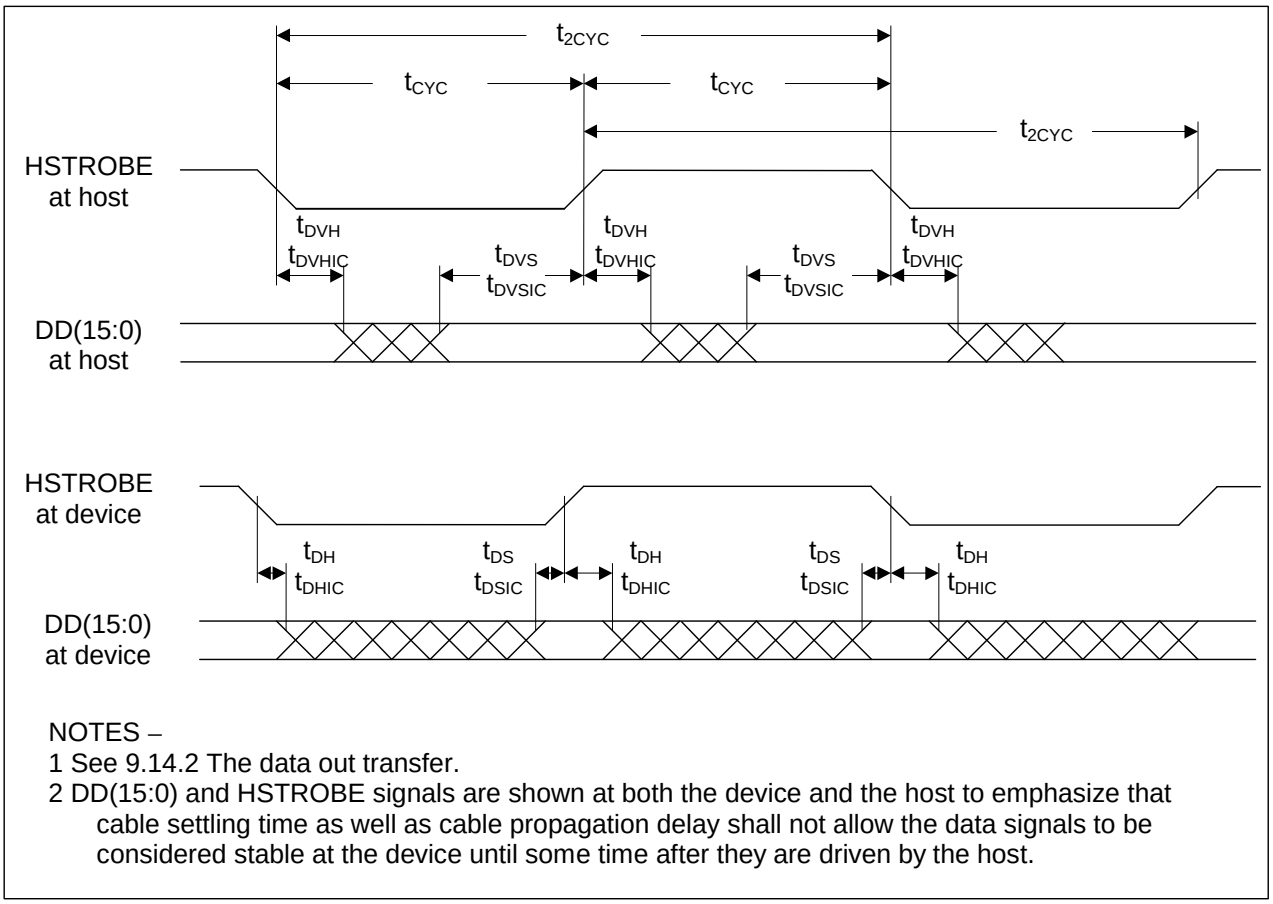



NOTES –
1 See 9.14.1 Initiating an Ultra DMA data out burst.
2 The definitions for the STOP, DDMARDY, and HSTROBE signal lines are not in effect until DMARQ and DMACK are asserted.

**Figure 76 – Initiating an Ultra DMA data-out burst**

#### 12.2.4.7 Sustained Ultra DMA data-out burst

The values for the timings for each of the Ultra DMA modes are contained in 12.2.4.

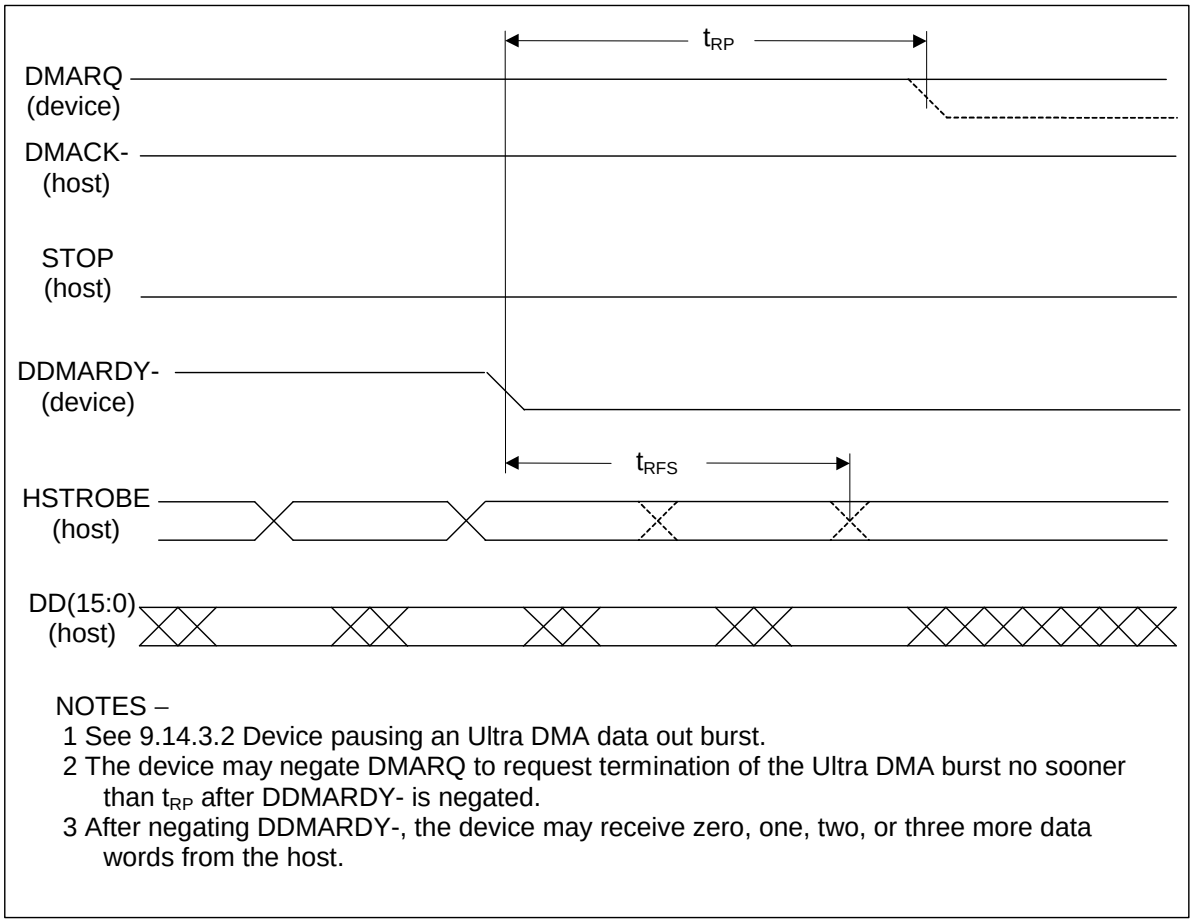



NOTES –
1 See 9.14.2 The data out transfer.
2 DD(15:0) and HSTROBE signals are shown at both the device and the host to emphasize that cable settling time as well as cable propagation delay shall not allow the data signals to be considered stable at the device until some time after they are driven by the host.

**Figure 77 – Sustained Ultra DMA data-out burst**

#### 12.2.4.8 Device pausing an Ultra DMA data-out burst

The values for the timings for each of the Ultra DMA modes are contained in 12.2.4.

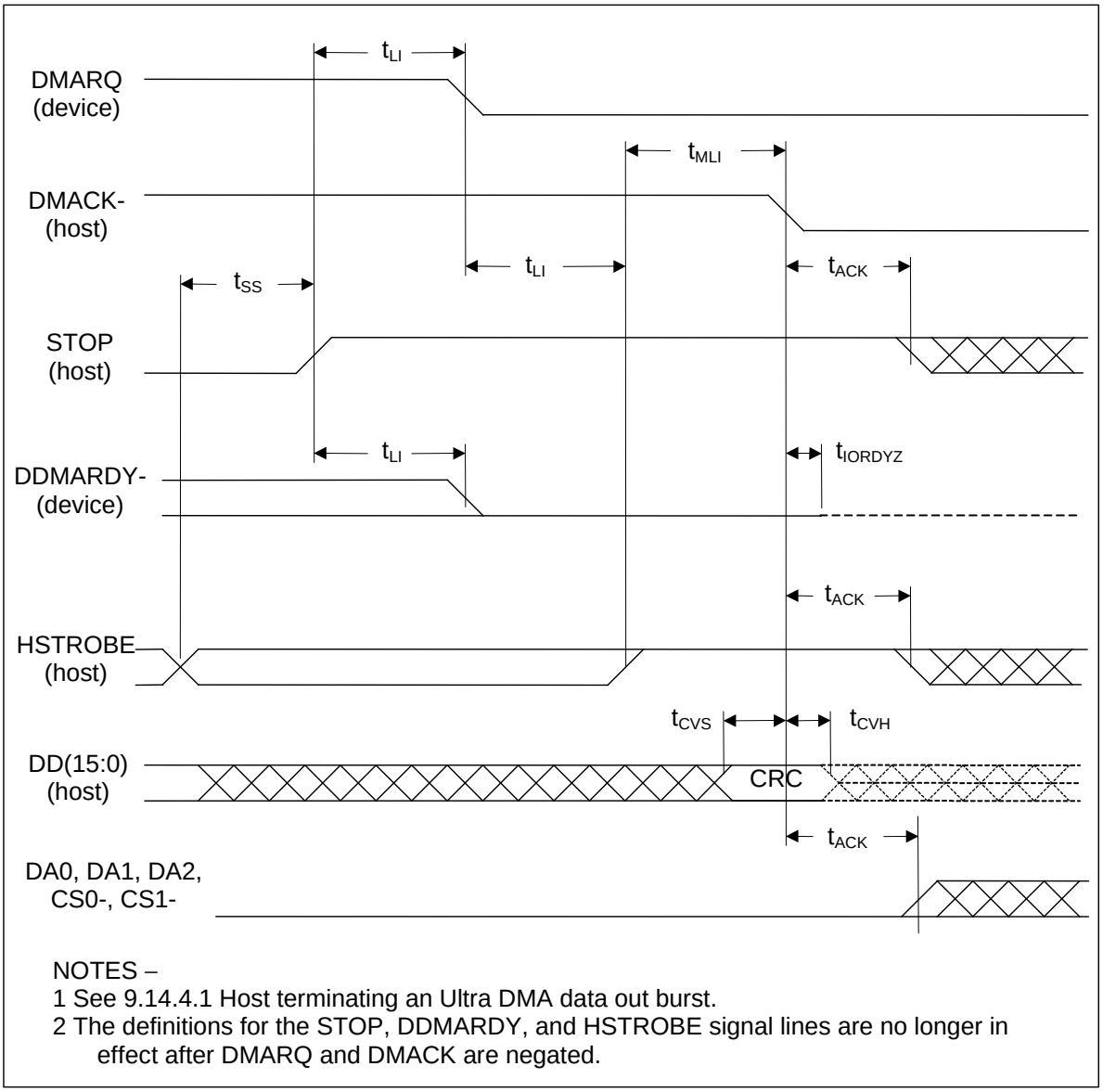



NOTES –
1 See 9.14.3.2 Device pausing an Ultra DMA data out burst.
2 The device may negate DMARQ to request termination of the Ultra DMA burst no sooner than $t_{RP}$ after DDMARDY- is negated.
3 After negating DDMARDY-, the device may receive zero, one, two, or three more data words from the host.

**Figure 78 – Device pausing an Ultra DMA data-out burst**

#### 12.2.4.9 Host terminating an Ultra DMA data-out burst

The values for the timings for each of the Ultra DMA modes are contained in 12.2.4.

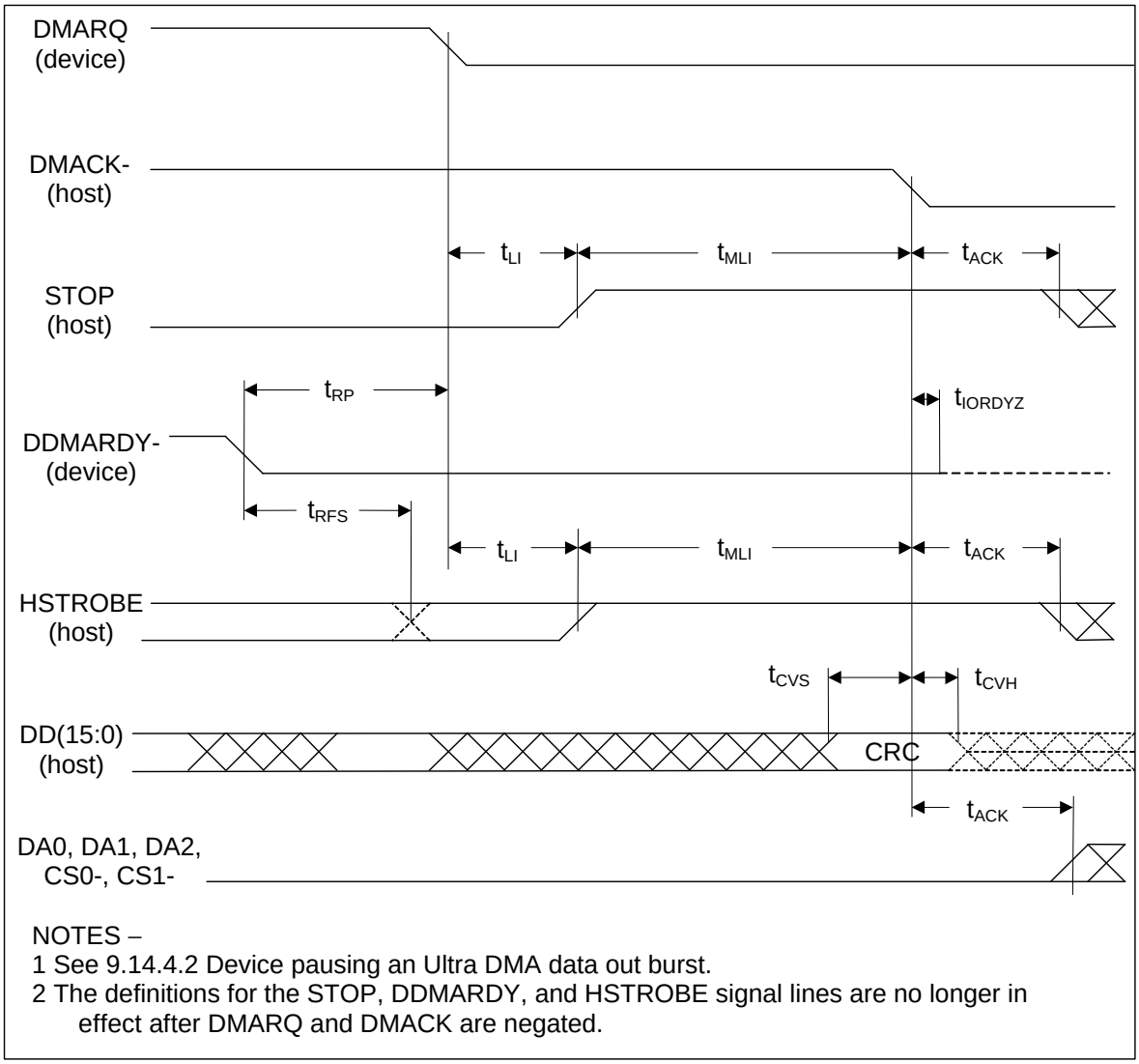



NOTES –
1 See 9.14.4.1 Host terminating an Ultra DMA data out burst.
2 The definitions for the STOP, DDMARDY, and HSTROBE signal lines are no longer in effect after DMARQ and DMACK are negated.

**Figure 79 – Host terminating an Ultra DMA data-out burst**

#### 12.2.4.10 Device terminating an Ultra DMA data-out burst

The values for the timings for each of the Ultra DMA modes are contained in 12.2.4.




NOTES –
1 See 9.14.4.2 Device pausing an Ultra DMA data out burst.
2 The definitions for the STOP, DDMARDY, and HSTROBE signal lines are no longer in effect after DMARQ and DMACK are negated.

**Figure 80 – Device terminating an Ultra DMA data-out burst**

**13 Serial interface overview** (See Volume 3)

**14 Serial interface physical layer** (See Volume 3)

**15 Serial interface link layer** (See Volume 3)

**16 Serial interface transport layer** (See Volume 3)

**17 Serial interface device command layer** (See Volume 3)

**18 Serial interface host adapter interface** (See Volume 3)

**19 Serial interface error handling** (See Volume 3)

# Annex A
(informative)
# Bibliography

AT Attachment with Packet Interface (ATA/ATAPI-5), ANSI NCITS.340-2000
AT Attachment with Packet Interface (ATA/ATAPI-6), ANSI INCITS 361-2002
BIOS Enhanced Disk Drive Specification (EDD), NCITS TR-21:1998
BIOS Enhanced Disk Drive Services, ANSI NCITS.347-2001
BIOS Enhanced Disk Drive Services - 2, ANSI NCITS.363-200n
Address Offset Reserved Area Boot, NCITS TR-27:2001
A.X. Widmer and P.A. Franaszek, “A DC-Ballanced, Prtitioned-Block, 8b/10b Transmission Code”. IBM Journal of Research and Development, 27, no. 5: 440-451 (September, 1983)
U.S. Patent 4,486,739. Peter A. Franaszek and Albert X. Widmer. Byte Oriented DC Balanced (0,4) 8b/10b Partitioned Block Transmission Code. (December 4, 1984)

# Annex B
(informative)
# Command set summary

(See Volume 1)

# Annex C
(informative)
# Design and programming considerations for large physical sector devices

(See Volume 1)

# Annex D
(normative)
# Device determination of cable type

## D.1 Overview

This standard requires that, for systems using a cable assembly, an 80-conductor cable assembly shall be installed before a system may operate with Ultra DMA modes greater than 2. However, some hosts have not implemented circuitry to determine the installed cable type by detecting whether PDIAG-:CBLID- is connected to ground as mandated by this standard. The following describes an alternate method for using IDENTIFY DEVICE or IDENTIFY PACKET DEVICE data from the device to determine the cable type. It is not recommended that a host use the method described in this annex.

If a host uses IDENTIFY DEVICE or IDENTIFY PACKET DEVICE data from the device to determine the cable type, then a 0.047 μf capacitor shall be installed from CBLID- to ground at the host connector. The tolerance on this capacitor is +/- 20% or less. After receiving an IDENTIFY DEVICE or IDENTIFY PACKET DEVICE command the device detects the presence or absence of the capacitor by asserting PDIAG-:CBLID- to discharge the capacitor, releasing PDIAG-, and sampling PDIAG-:CBLID- before the installed capacitor could recharge through the 10 kΩ pull-up resistor(s) on PDIAG-:CBLID- at the device(s).

If the host system has a capacitor on PDIAG-:CBLID- and a 40-conductor cable is installed, the rise time of the signal will be slow enough that the device will sample PDIAG-:CBLID- while the signal is still below $V_{IL}$. Otherwise, if PDIAG-:CBLID- is not connected from the host connector to the devices in an 80-conductor cable assembly, the device will detect that the signal is pulled above $V_{IH}$ through the resistor(s) on the device(s). The capacitor test results will then be reported to the host in the IDENTIFY DEVICE or IDENTIFY PACKET DEVICE data. The host will use the data to determine the maximum transfer rate of which the system is capable and use this information when setting the transfer rate using the SET FEATURES command.

## D.2 Sequence for device detection of installed capacitor

The following is the sequence for a host using IDENTIFY DEVICE or IDENTIFY PACKET DEVICE data from the device to determine the cable type:

a) the host issues an IDENTIFY DEVICE or IDENTIFY PACKET DEVICE command (according to device type) first to Device 1 and then to Device 0 after every power-on or hardware reset sequence (the command is issued to Device 1 first to ensure that Device 1 releases PDIAG-:CBLID- before Device 0 is selected. Device 0 will be unable to distinguish a discharged capacitor if Device 1 is driving the line to its electrically low state. Issuing the command to Device 1 forces it to release PDIAG-:CBLID-);
b) the selected device asserts PDIAG-:CBLID- for at least 30 μs after receipt of the IDENTIFY DEVICE or IDENTIFY PACKET DEVICE command but before transferring data for the command;
c) the device releases PDIAG-:CBLID- and samples it between two and thirteen μs after release;
d) if the device detects that PDIAG-:CBLID- is below $V_{IL}$, then the device returns a value of zero in bit 13 of word 93 in its IDENTIFY DEVICE or IDENTIFY PACKET DEVICE data (if the host system has a capacitor on that signal and a 40-conductor cable is installed, the rise time of the signal will be slow enough that it will be sampled by the device while it is still below $V_{IL}$);
e) if the device detects that the signal is above $V_{IH}$, then the device returns a value of one in bit 13 of word 93 in its IDENTIFY DEVICE or IDENTIFY PACKET DEVICE data. This signal is not connected between the host and the devices in an 80-conductor cable assembly, thus, the sampling device will see this signal pulled above $V_{IH}$ through the 10 kΩ resistor(s) installed on the device(s);
f) the host then uses its knowledge of its own capabilities and the content of word 88 and word 93 to determine the Ultra DMA modes of which the system is capable;
g) the host then uses the SET FEATURES command to set the transfer mode.

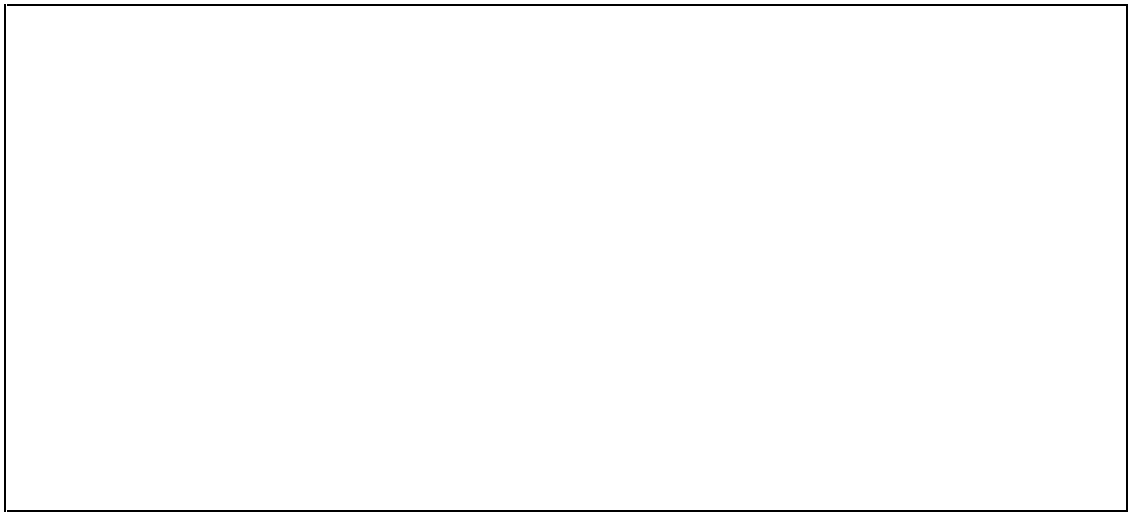


**Figure D.1 - Example configuration of a system where the device detects a 40-conductor cable**

**Table D.1 - Device detection of installed capacitor**

| Cable assembly type | Device 1 releases PDIAG- | Value reported in ID data by device | Device-determined cable type | Determination correct? |
|---|---|---|---|---|
| 40-conductor | Yes | 0 | 40-conductor | Yes |
| 80-conductor | Yes | 1 | 80-conductor | Yes |
| 40-conductor | No | 0 | 40-conductor | Yes |
| 80-conductor | No | 0 | 40-conductor | No (see note) |
| NOTE – Ultra DMA modes greater than 2 will not be set even though the system supports them. | | | | |

**Table D.2 - Results of device based cable detection if the host does not have the capacitor installed**

| Cable assembly type | Device 1 releases PDIAG- | Value reported in ID data by device | Device-determined cable type | Determination correct? |
|---|---|---|---|---|
| 40-conductor | Yes | 1 | 80-conductor | No (see note 1) |
| 80-conductor | Yes | 1 | 80-conductor | Yes |
| 40-conductor | No | 0 | 40-conductor | Yes |
| 80-conductor | No | 0 | 40-conductor | No (see note 2) |
| NOTES -<br>1 Ultra DMA modes greater than 2 may be set incorrectly resulting in ICRC errors.<br>2 Ultra DMA modes greater than 2 will not be set even though the system supports them. | | | | |

### D.3 Using the combination of methods for detecting cable type

Determining the cable assembly type may be done either by the host sensing the condition of the PDIAG-:CBLID- signal, by relying on information from the device, or a combination of both methods. Table D.3 describes the results of using both host and device cable detection methods.

**Table D.3 - Results of using both host and device cable detection methods**

| Cable assembly type | Device 1 Releases PDIAG- | Electrical state of CBLID- at host | Value reported in ID data by device | Determined cable type | Determination correct? |
|---|---|---|---|---|---|
| 40-conductor | Yes | 1 | 0 | 40 | Yes |
| 80-conductor | Yes | 0 | 1 | 80 | Yes |
| 40-conductor | No | 0 | 0 | 40 | Yes (see note) |
| 80-conductor | No | 0 | 0 | 40 | No (see note) |

NOTE - The 0,0 result is independent of cable type and indicates that Device 1 is incorrectly asserting PDIAG-. When the host determines this result, it shall not operate with Ultra DMA modes greater than 2 and it may respond in several ways:

1 report that Device 1 is incompatible with Ultra DMA modes higher than 2 and should be used on a different port in order to use those modes on the port being detected;

2 report that Device 1 is not allowing the cable type to be properly detected;

3 do not notify the user of any problem but detect the cable as a 40-conductor.

The Table D.4 below illustrates intermediate results for all combinations of cable, device, and host, for hosts that support Ultra DMA modes greater than 2.

**Table D.4 - Results for all combinations of device and host cable detection methods**

| Design options | | | Intermediate actions and results | | | | | | Results |
|---|---|---|---|---|---|---|---|---|---|
| 80-conductor cable installed | Device supports UDMA modes >2 | Host senses PDIAG-: CBLID- | Host uses ID data, capacitor installed | Host capacitor connected to device | Device tests for capacitor | Capacitor detected | ID word 93 Bit 13 value | Host checks ID word 93 bit 13 | Host may set UDMA mode >2 |
| No | No | Yes | No | No | No | No | 0 | No | No |
| No | Yes | Yes | No | No | Yes | No | 1 | No | No |
| Yes | No | Yes | No | No | No | No | 0 | No | No |
| Yes | Yes | Yes | No | No | Yes | No | 1 | No | Yes |
| No | No | No | Yes | Yes | No | No | 0 | Yes | No |
| No | Yes | No | Yes | Yes | Yes | Yes | 0 | Yes | No |
| Yes | No | No | Yes | No | No | No | 0 | Yes | No |
| Yes | Yes | No | Yes | No | Yes | No | 1 | Yes | Yes |

# Annex E
(informative)
# Signal integrity and UDMA implementation guide

## E.1 Introduction

This annex is intended as an aid to the implementation of Ultra DMA in host systems, ATA controllers, and peripherals. Clarification of some aspects of the protocol and details not specifically stated in the normative sections of the standard have been included for the benefit of component, PCB, and device driver engineers. This annex is not intended to be comprehensive but rather informative on subjects that have caused design questions. Included are warnings about proper interpretation of protocol where interpretation errors seem possible. The information provided is relevant to implementation of all Ultra DMA modes 0 through 6, as well as earlier protocols.

This annex uses the term data-out to indicate a transfer from the host to a device and data-in to indicate a transfer from the device to the host.

The ATA bus is a storage interface originally designed for the ISA Bus of the IBM PC/AT™. With the advent of faster host systems and devices, the definition of the bus has been expanded to include new operating modes. Each of the PIO modes, numbered zero through four, is faster than the one before (higher numbers translate to faster transfer rates). PIO modes 0, 1, and 2 correspond to transfer rates for the interface as was originally defined with maximum transfer rates of 3.3, 5.2, and 8.3 megabytes per second (MB/s), respectively. PIO mode 3 defines a maximum transfer rate of 11.1 MB/s, and PIO mode 4 defines a maximum rate of 16.7 MB/s. Additionally, Multiword DMA and Ultra DMA modes have been defined. Multiword DMA mode 0, 1, and 2 have maximum transfer rates of 4.2, 13.3, and 16.7 MB/s, respectively. Ultra DMA modes 0, 1, 2, 3, 4, and 5 have maximum transfer rates of 16.7, 25, 33.3, 44.4, 66.7, 100 and 133 MB/s, respectively.

Ultra DMA features such as increased frequencies, double-edge clocking, and non-interlocked signaling require improved signal integrity on the bus relative to that required by PIO and Multiword DMA modes. For Ultra DMA modes 0, 1, and 2 this is achieved by the use of partial series termination and controlled slew rates. For modes 3 and above an 80-conductor cable assembly is required in addition to partial series termination and controlled slew rates. This cable assembly has ground lines between all signal lines on the bus in order to control impedance and reduce crosstalk, eliminating many of the signal integrity problems inherent to the 40-conductor cable assembly. However, many of the design considerations and measurement techniques required for the 80-conductor cable assembly are different from those used for the 40-conductor assembly. Hosts and devices capable of Ultra DMA modes greater than 2 should be designed to meet all requirements for operation with both cable types. Unless otherwise stated, 40- and 80-conductor cables are assumed to be 18 inches long, the maximum allowed by this standard. Timing and signal integrity issues as discussed apply to this length cable.

## E.2 The issues

The following describe the issues and design challenges while providing suggestions for implementation with respect to timing, crosstalk, ground bounce, and ringing.

### E.2.1 Timing

Two of the features Ultra DMA introduced to the bus are double-edge clocking and non-interlocked (also known as source-synchronous) signaling. Double-edge clocking allows a word of data to be transferred on each edge of STROBE (this is HSTROBE for an Ultra DMA data-out transfer and DSTROBE for a data-in transfer), resulting in doubling the data rate without increasing the fundamental frequency of signaling on the bus. Non-interlocked signaling means that DATA and STROBE are both generated by the sender during a data transfer. In addition to signal integrity issues such as clocking the same data twice due to ringing on the

STROBE signal and delay-limited interlock timings on the bus, non-interlocked signaling makes settling time and skew between different signals on the bus critical for proper Ultra DMA operation.

#### E.2.1.1 Cabling

The 80-conductor cable assembly adds 40 ground lines to the cable between the 40 signal lines defined for the 40-conductor cable assembly. These added ground lines are connected inside each connector on the cable assembly to the seven ground pins defined for the 40-conductor cable assembly. These additional ground lines allow the return current for each signal line to follow a closer path to the outgoing current than was allowed by the grounding scheme in the 40-conductor cable assembly. This results in a lower impedance and greatly reduced crosstalk for signals on the data bus. The controlled impedance and reduced crosstalk of the 80-conductor cable assembly results in much improved behavior of electrical signals on the bus and reduces the data settling time to effectively zero regardless of switching conditions. Thus, the signal at the recipient is monotonic, such that the first crossing of the input threshold is considered final. Reducing the time allowed for data settling time (DST) from greater than 25 ns in Ultra DMA mode 2, to 0 ns with the 80-conductor cable assembly allows nominal cycle time to be reduced from 60 ns for mode 2, to 15 ns for mode 6.

#### E.2.1.2 Skew

Skew is the difference in total propagation delay between two signals as they transit the bus. Propagation delay is the amount of time required for a single input signal at one part of the system to cause a disturbance to be observed at another part of the system in a system containing continuously distributed capacitance and inductance. Propagation delay is determined by the velocity of light within the dielectric materials containing the electric fields in the system. For systems with uniform properties along their length, propagation delay is often specified as seconds per foot or seconds per meter.

Skew will be positive or negative depending on which signal is chosen as the reference. All skews in the Ultra DMA timing derivations are defined as STROBE delay minus data delay. A positive skew is a STROBE that is delayed more than the data.

Skew corresponds to the reduction in setup and hold times that occurs between the sender and the recipient. If the bus contributes skew that exceeds the difference between the setup time produced by the sender and that required by the recipient, data will be stored incorrectly. The same is true for hold time. Skew between signals is caused by differences in the electrical characteristics of the paths followed by each signal.

Ultra DMA modes higher than 4 require less skew within the physical cable system than lowermodes. In order to reduce the amount of skew created as signals transit the system, modes higher than 4 place a number of new requirements on the analog electrical aspects of system design. The primary requirement is that all devices and hosts supporting modes higher than 4 use 3.3 volt signaling. This eliminates the contribution to skew from the asymmetry of the input thresholds with the previous 5 volt $V_{oH}$. A second requirement is that hosts use a 4.7 kΩ pull-up resistor on IORDY/DSTROBE instead of the 1 kΩ resistor used for previous modes. The pull-up shall be to the host's 3.3 V internal supply. Third, the total output impedance consisting of driver resistance plus series termination resistor shall match the typical cable impedance of 75 to 85 Ω.

#### E.2.1.3 Source-terminated bus

The bus operates as a source-terminated bus, meaning that the only low-impedance connection to ground is via the source impedance of the drivers in the sender.

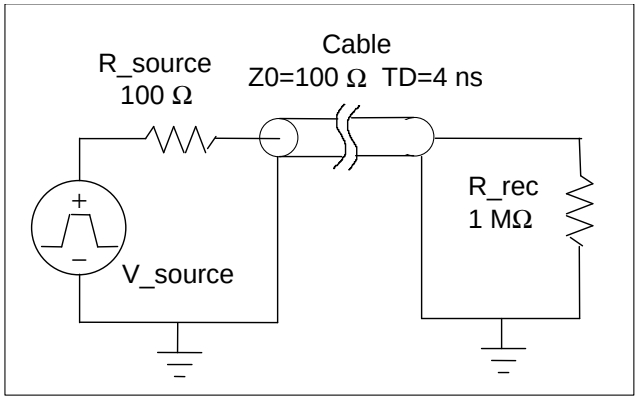



**Figure E.1 – A transmission line with perfect source termination**

On a source-terminated transmission line, the initial voltage level produced at the source propagates through the system until it reaches the receiving end that, by definition, is an open circuit or at least has high impedance relative to the characteristic impedance of the transmission line. This open circuit produces a reflection of the original step with the same polarity and amplitude as the original step but travelling in the opposite direction. The reflected step adds to the first step to raise the voltage throughout the system to two times the original step voltage. In a perfectly terminated system (see figure E.1), R_source matches the cable impedance resulting in an initial step voltage on the transmission line equal to fifty percent of V_source, and the entire system has reached a steady state at V_source once the reflection returns to the source.

The waveforms that are measured on the bus as a result of this behavior depend on the ratio of the signal rise time to the propagation delay of the system. If the rise time is shorter than the one-way propagation delay, the initial voltage step will be visible at the sender. At the recipient the incoming voltage step is instantaneously doubled as it reflects back to the sender and no step is observed (see figure E.2).

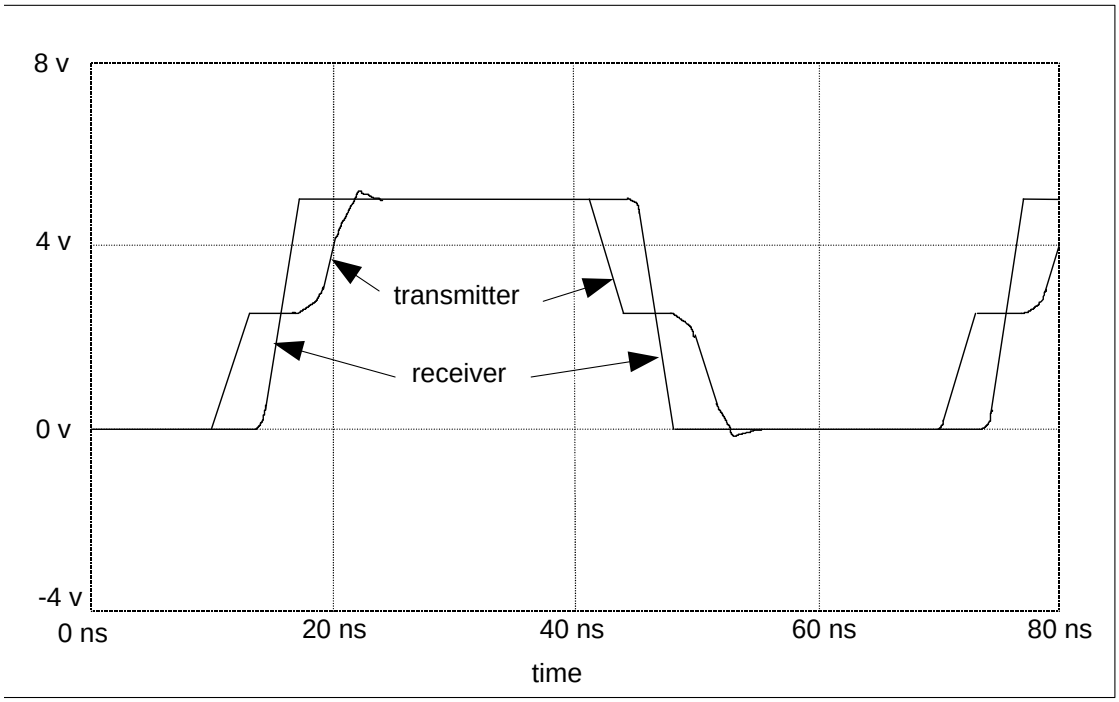



**Figure E.2 - Waveforms on a source-terminated bus with rise time less than $T_{prop}$**

If the rise time is longer than the propagation delay, the sender waveform changes, but the same behavior still occurs: the reflected step adds to the initial step at the sender while a delayed doubling of the initial step is observed at the recipient. Because the rising edges of the two steps overlap when measured at the sender, there is a temporary increase in slew rate instead of a step seen at the sender while the rising edge of the reflection adds to the edge still being generated by the sender (see figure E.3).

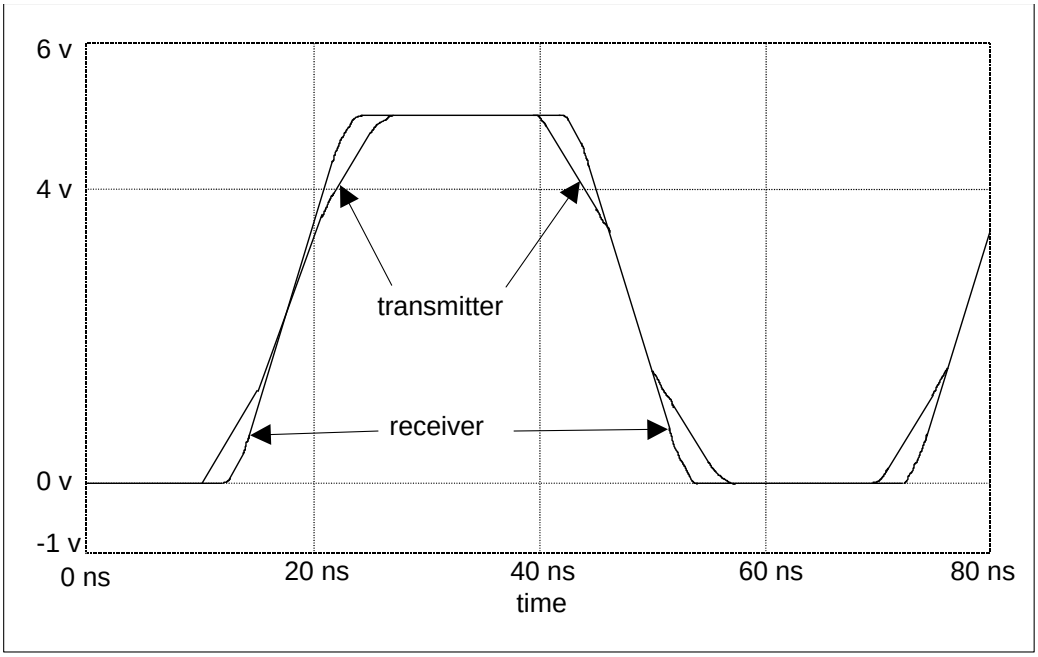



**Figure E.3 - Waveforms on a source-terminated bus with rise time greater than $T_{prop}$**

In figure E.2 and E.3, the source impedance is perfectly matched to the cable impedance with the result that, after the first reflection returns to the source, there are no further reflections, and the system is at a steady state. In a system that is not perfectly terminated, there are two possibilities. The first possibility is when the source impedance is less than the characteristic impedance of the transmission line, the initial step is greater than fifty percent of $V_{oH}$, and the system is at a voltage higher than $V_{oH}$ when the first reflection returns to the recipient (see figure E.4). In this case another reflection occurs at the source to reduce the system to a voltage below $V_{oH}$ but closer to $V_{oH}$ than the initial peak. Reflections continue but are further reduced in amplitude each time they reflect from the termination at the source.

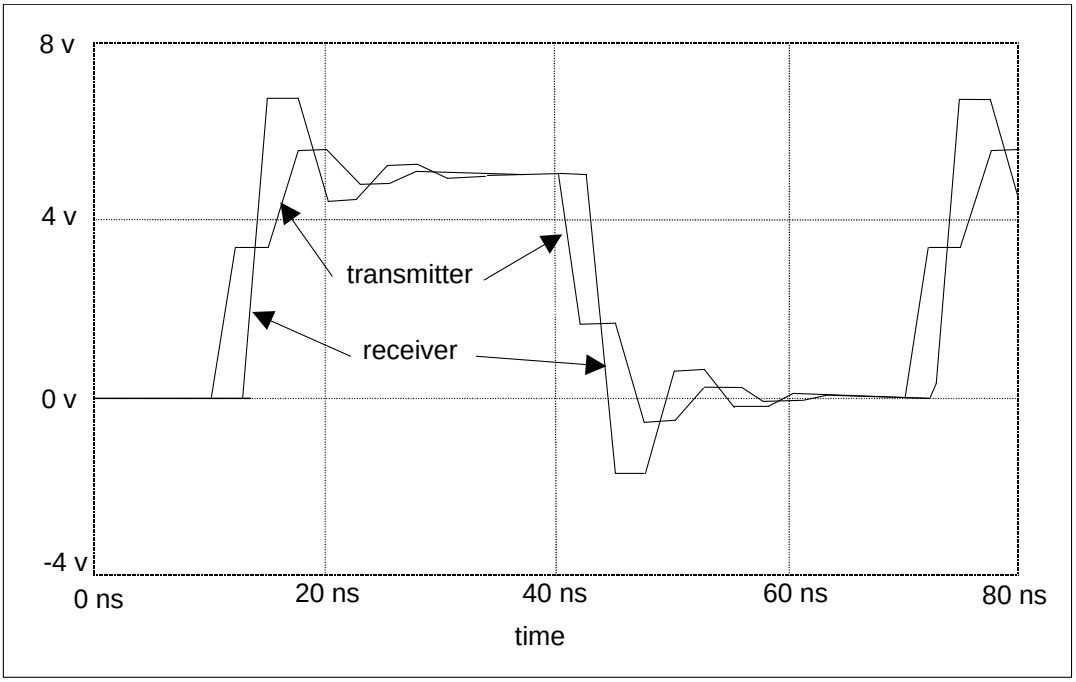



**Figure E.4 - Waveforms on a source-terminated bus with R_source less than cable $Z_0$**

The second possibility is when the source impedance is higher than the characteristic impedance, the initial step is less than fifty percent of $V_{oH}$, and multiple reflections back and forth on the bus will be required to bring the whole system up to a steady state at $V_{oH}$ (see figure E.5).

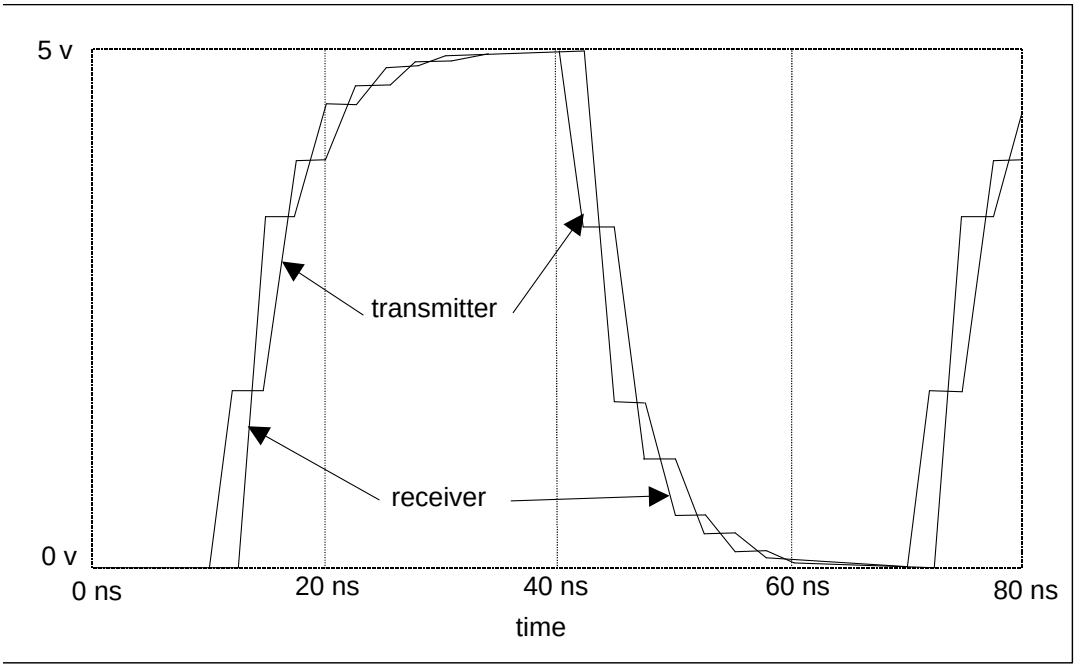



**Figure E.5 - Waveforms on a source-terminated bus with R_source greater than cable $Z_0$**

Note that falling edges exhibit the same transmission line behavior as rising edges. The only difference between the edges is that $V_{oH}$ and $V_{oL}$ are reversed. In actual systems output impedance and slew rate of the drivers are often different between rising and falling edges, resulting in different step voltages and waveform shapes.

For typical implementations using 33 Ω series termination, the effective driving impedance of a sender's component I/O viewed from the cable connector ranges from 50 to 90 Ω. The component I/O is the combined input and/or output circuitry, bond wire, and pin on an IC that is responsible for receiving and/or sending data on a particular conductor within the bus. The initial voltage step produced when an edge is driven onto the cable will be equal to the driver's open-circuit $V_{oH}$ divided by the effective output impedance and the input impedance of the cable (typically 82 Ω), or a 50 to 60 Ω printed circuit board trace in the case of hosts. This step voltage will fall in the range from 50 to 70 percent of $V_{oH.}$ For example, for a theoretical source with zero output impedance using 33 Ω termination driving an 82 Ω cable the resulting step voltage is not greater than $100 * ( 82 \div (33 + 82 ) )$ = 71.3 percent of $V_{oH}$. Because the thresholds of an input are not centered with respect to the high and low voltages, the initial voltage step produced by a driver will often cross the recipient's input threshold on a rising edge but not on a falling edge. However, since the signal received at the end of the bus is a doubled version of the initial output from the sender, it will cross the switching thresholds for any reasonably low output impedance. Because of this the main voltage step only affects skew and delay for signals received at devices that are not at the end of the cable. The greater the distance a device is from the device end of the cable (i.e., closer to the host), the longer the duration of the step observed (see figures E.6 and E.7).

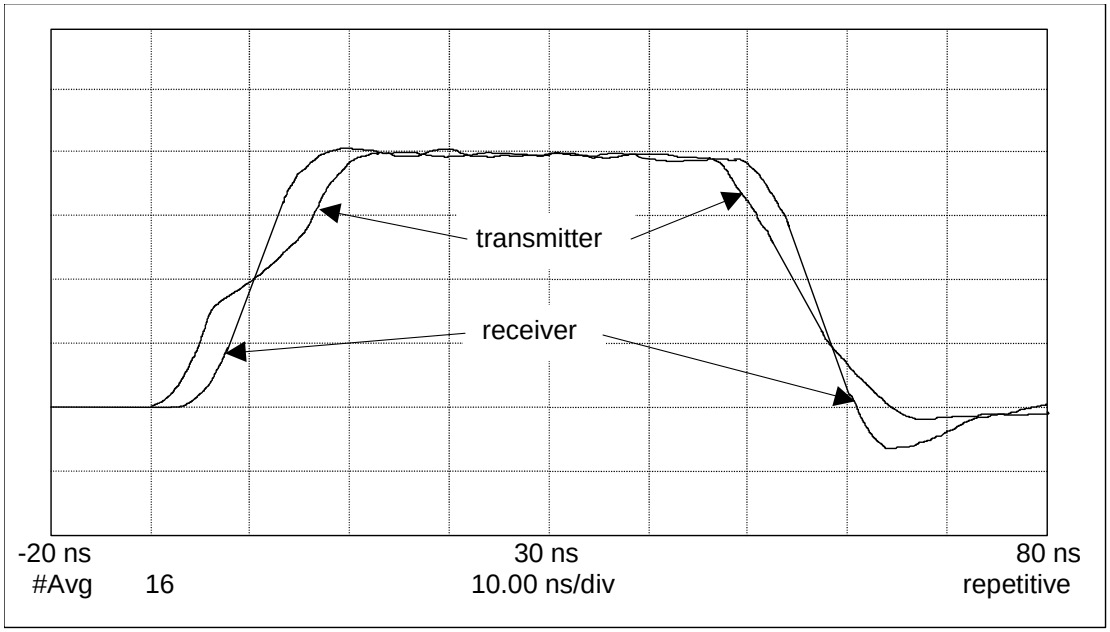



**Figure E.6 - Typical step voltage seen in ATA systems using an 80-conductor cable (measured at drive and host connectors during read)**

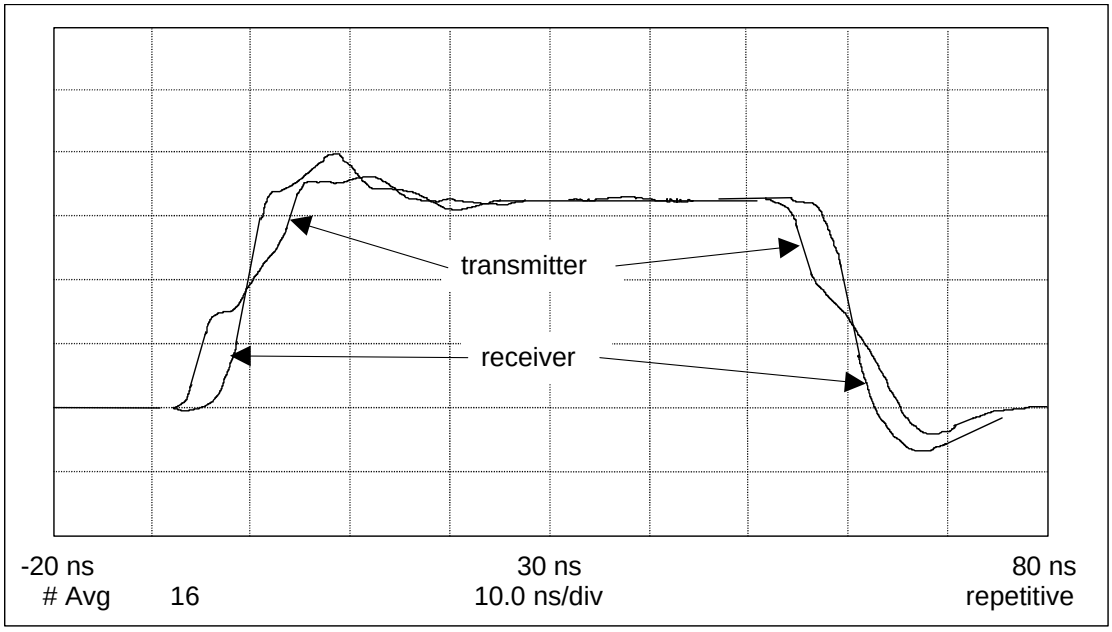



**Figure E.7 - Typical step voltage seen in ATA systems using an 80-conductor cable (measured at host and drive connectors during write)**

In addition to the step produced by the initial voltage driven onto the bus and the subsequent reflection, smaller steps are produced each time the propagating signal encounters a change in the bus impedance. The major impedance changes that occur in a system are: 1) at the connections between the cable and the printed circuit boards (PCBs) of the hosts and devices, 2) along the traces of the PCBs as the result of changing layers, and 3) at the connection between a motherboard and a backplane.

The transmission line behavior of the 80-conductor cable assembly adds skew to the received signal in two ways. First, impedance differences along one line versus another will result in different amounts of delay and attenuation on each line due to reflections on the bus. This produces a time difference between the two signals' threshold crossings at the recipient. Secondly, signals received at the device that is not at the end of the cable may cross the threshold during the initial voltage step or after the reflection from the end of the

cable is received, depending on the supply voltage, series termination, output impedance, $V_{oH}$, and PCB trace characteristics of the host.

Factors other than cable characteristics also contribute to skew. Differences in the capacitive loading between the STROBE and DATA lines on devices attached to the bus will delay propagating signals by differing amounts. Differences in slew rate or output impedance between drivers when driving the 82 Ω load will result in skew being generated as the signal is sent at the sender. Differences between the input RC delays on STROBE and DATA lines will add skew at the recipient.

The fundamental requirement for minimizing skew in the entire system is to make the STROBE and DATA lines as uniform as possible throughout the system.

#### E.2.1.4 Timing measurements for the 80-conductor cable assembly

The reflections that are present in a system make it difficult to measure skew and delays accurately. For the received signal at a device, the propagation delay from the device connector to the device integrated circuit (IC) connector pin is about 300 ps for typical device PCBs and trace lengths. The IC is the entire component (die and package) that contains the ATA bus interface circuitry.

This delay introduces an error of plus or minus 300 ps in timing measurements made at the device connector since rising edges and falling edges will be measured before and after the step respectively. When comparing two signals, this results in an error in measured skew of plus or minus 600 ps due to the measurement position. This error is small enough relative to the total timing margin of an Ultra DMA system that it may be ignored in most cases.

Since the trace length on host PCBs are often much longer than those on devices, the propagation time for a signal from the host connector to the host IC may be as high as 2 ns. This results in a plus or minus 2 ns accuracy in the measurement of a single signal and a plus or minus 4 ns accuracy for skew between two signals. These errors are not removed by adding or subtracting an allowance for PCB propagation delay depending on rising or falling edges because characteristics of the PCB and termination will affect the step levels and skew that occur at the component I/Os. As a result of this, accurate measurements of skew in signals received at the host are made either at pins of the host IC, or at points on the PCB traces as close to the IC pins as possible. Test pads, headers, or unconnected vias in PCB layouts may be designed allowing connection to DATA, STROBE, and ground for this purpose.

It is important to note that the timing specifications for Ultra DMA in the standard are based on measuring signals at the interface connector.

#### E.2.1.5 Simulations for the 80-conductor cable assembly

The difficult nature of measuring skew in actual systems makes simulations a more important tool in determining the effect on skew of design decisions regarding component I/Os, PCB layout, cable lengths, and other aspects of system design. Because of the well-controlled impedance of the 80-conductor cable assembly, single line transmission line models provide accurate predictions of the delay through the bus based on a given design choice for a given set of conditions on the bus. To be certain of the system-wide consequences of particular design choices, a large number of simulations encompassing many different combinations of parameters were used to determine the timing specifications for Ultra DMA mode 5. Results of these simulations are also the basis of the guidelines that follow.

Output skew is measured at the connector of the sender into capacitive loads to ground of 15 pf and 40 pf. An alternate loading arrangement is to measure the signal produced at the end of an 18-inch 80-conductor cable assembly into typical device and host loads of 20 pf or 25 pf that are held uniform across STROBE and DATA lines. Skew is measured at the crossing of the 1.5 volt threshold. All combinations of rising and falling edges on the signals involved are used when skew is measured.

Minimizing output skew is the best assurance of reliable signaling across the full range of cable loading and recipient termination conditions that will occur in systems.

### E.2.2 Crosstalk

Although the ground-signal-ground configuration of the 80-conductor cable assembly greatly reduces coupling between wires on the cable, the host and device connectors generate a large amount of crosstalk because they still use the original ground configuration with no ground lines separating the 16 signals of the data bus. In addition, crosstalk between traces on the PCB may reach high levels in systems with long traces or with tight spacing between traces. Cumulative crosstalk plus ground bounce measured at the connector of the recipient in typical systems using the 80-conductor cable ranges from 400 mV to 1 V peak, in short pulses with a frequency content equivalent to the frequency content of the edge rates of the drivers being used. Although this level of total crosstalk may seem like a hazard to reliable signaling, crosstalk exceeding 800 mV detected at the recipient does not affect the setup or hold times when it occurs during the interval when other signals are switching (see figure E.8). This figure was generated using the first falling STROBE edge for a trigger and showing a middle data signal staying low while all other lines switch high to low. With infinite persistence, the pattern was then changed to all lines switching low to high for the same STROBE edge. The crosstalk that occurs on the line staying low while all others switch high to low is in excess of 800 mV but has more hold and setup time margin than data lines that are switching and therefore it does not reduce setup or hold time margin.

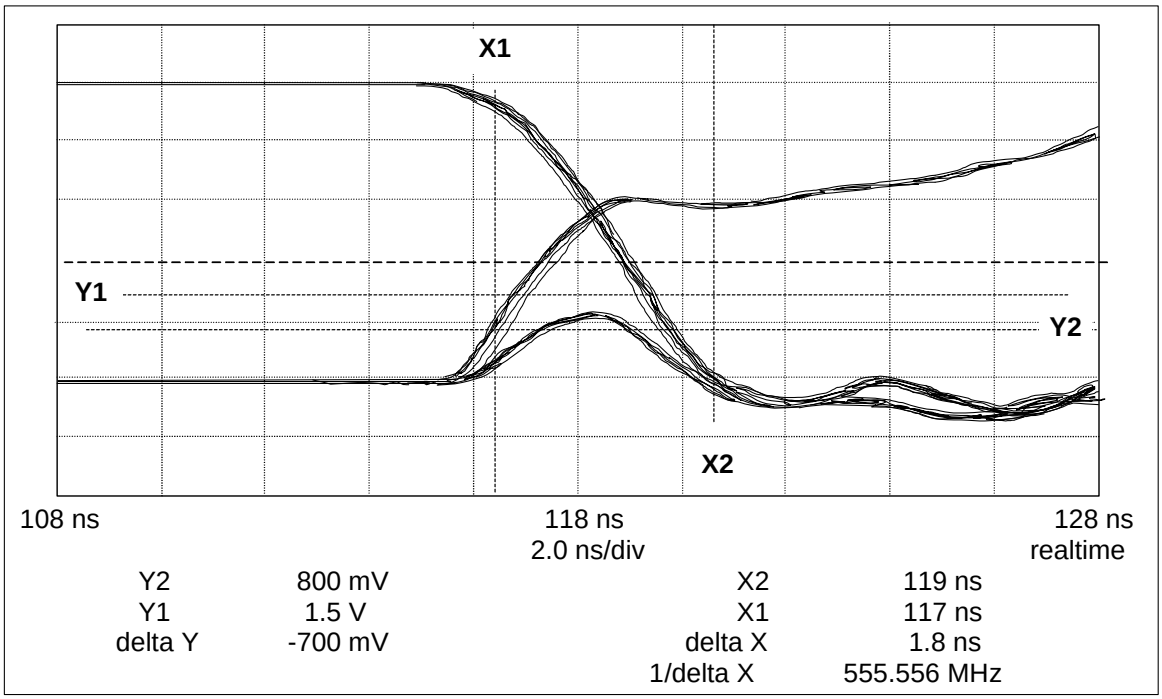



**Figure E.8 - Positive crosstalk pulse during a falling edge (does not affect data setup or hold time)**

A larger signal integrity hazard exists when crosstalk extends into the middle of the cycle when data could be clocked. This may result from a high level of reverse crosstalk detected at the recipient as the reflected signal propagates from the recipient input back to the sender output in the switching lines.

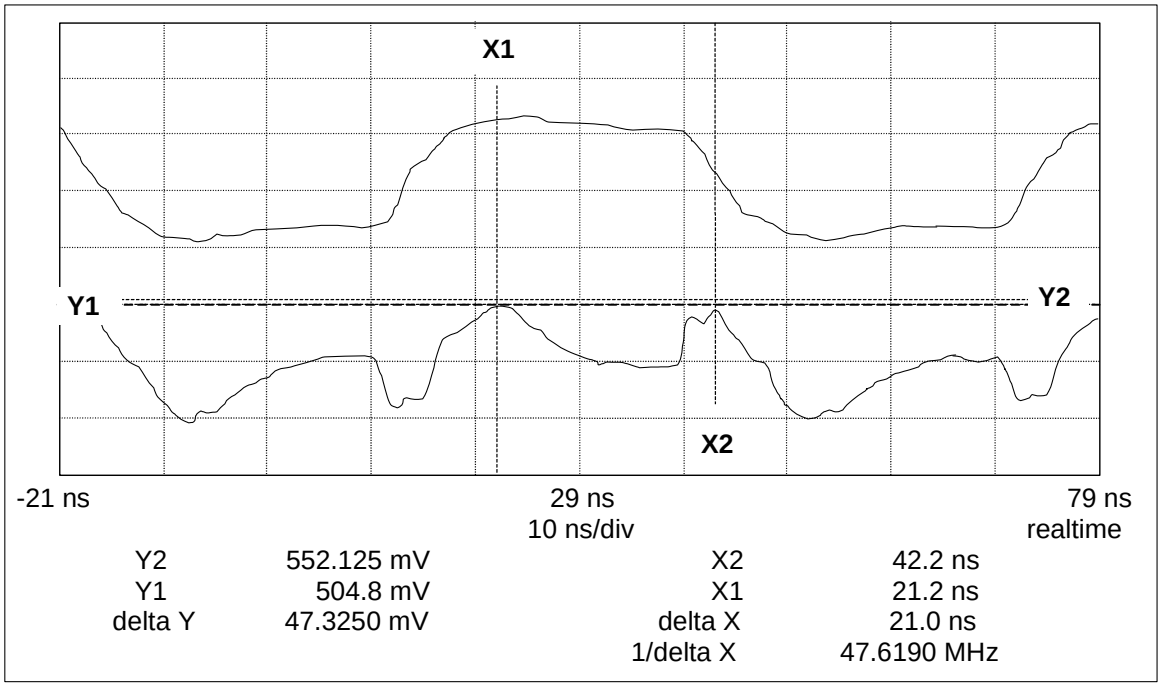



**Figure E.9 - Reverse crosstalk waveform from reflected edge (seen at the receiver in the middle of a cycle - marker X1)**

Reducing a system's creation of and susceptibility to forward and reverse crosstalk requires an understanding of how crosstalk is generated and propagates through the system. Crosstalk results from coupling between signals in the form of either a capacitance from one signal conductor to another or inductors in the path of each signal with overlapping magnetic fields. The capacitive and inductive coupling are easiest to understand if treated as separate effects.

### E.2.2.1 Capacitive coupling

Capacitive coupling in its simplest form consists of a capacitor connecting together two transmission lines somewhere along their length. When a change in voltage occurs on one line (called the aggressor line), a pulse on the non-switching signal (called the victim line) is produced with a peak amplitude proportional to the rate of change of voltage (dV/dt) on the aggressor line. The pulse on the victim line propagates both forward and backward from the point of coupling and has the same sign in both directions. Forward and backward are defined relative to the direction that the aggressor signal was propagating. Forward means that the propagation is in the same direction as the aggressor signal. Backward means that propagation is in the opposite the direction of the aggressor signal. Figure E.10 is a schematic of a model for capacitive coupling. Figure E.11 shows waveforms resulting from capacitive coupling at the sender and recipient component I/Os of the aggressor and victim lines.

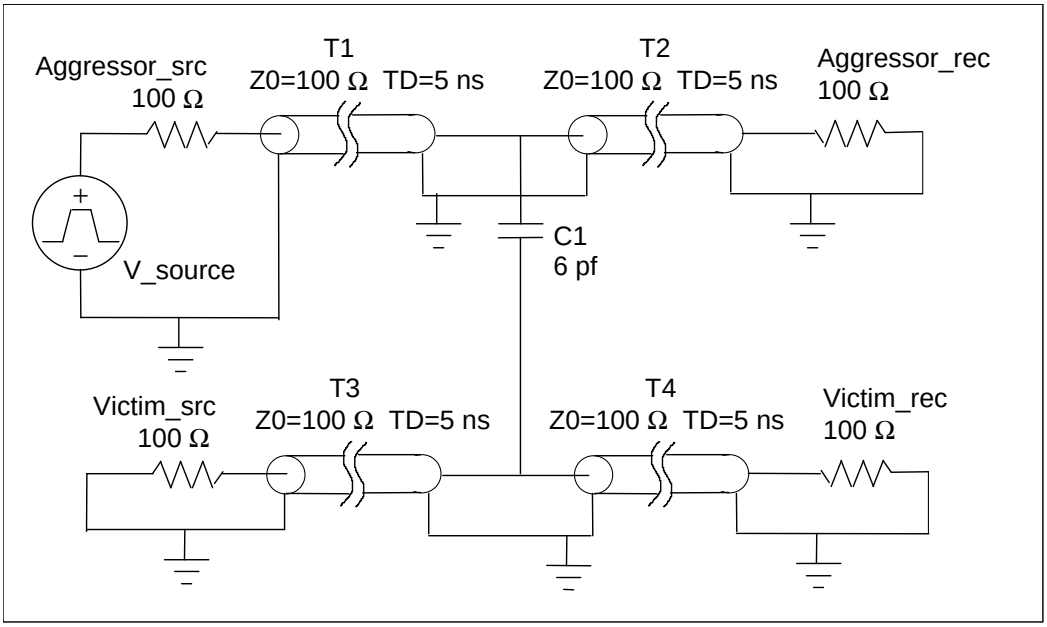



**Figure E.10 - Model of capacitive coupling**

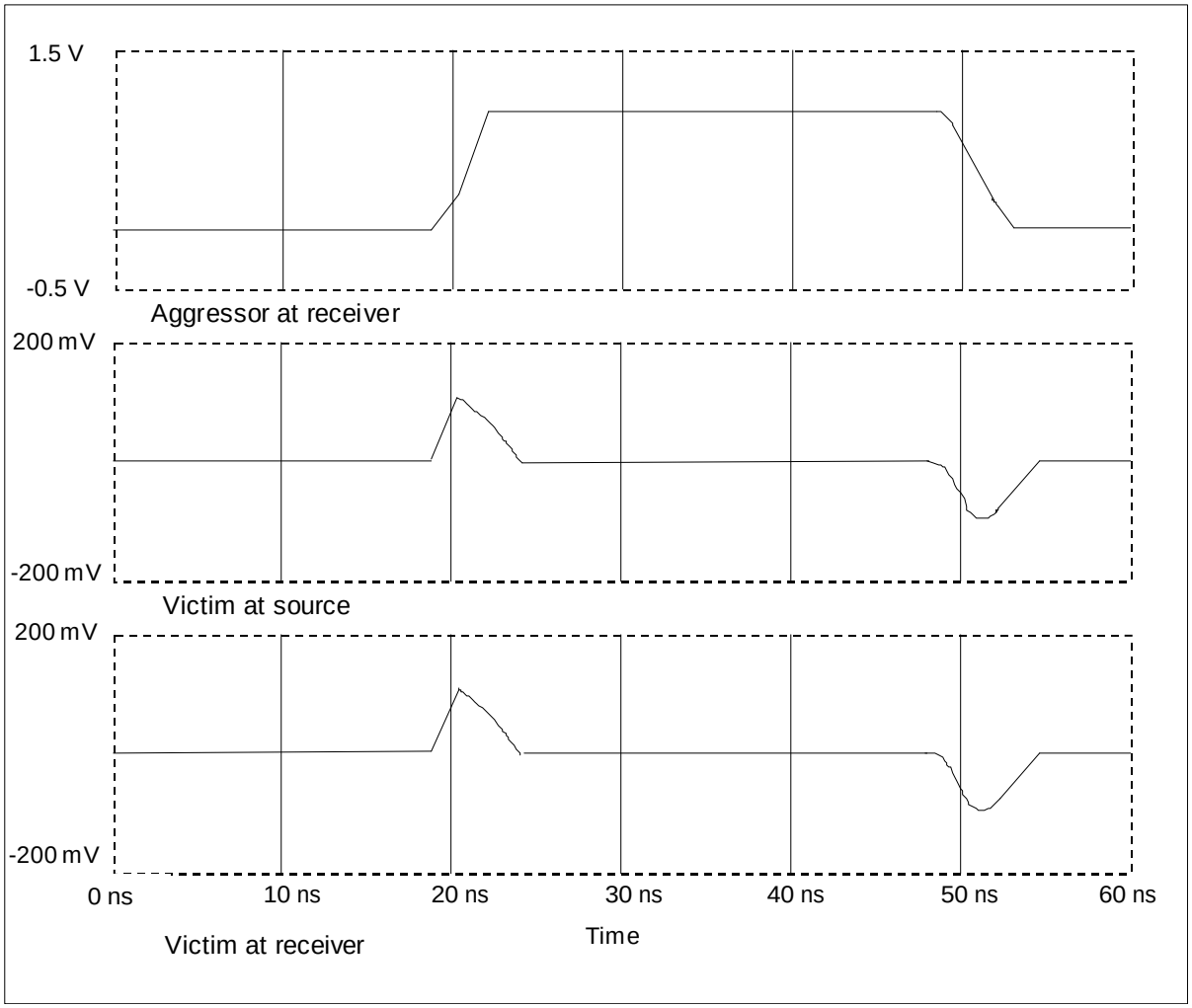



**Figure E.11 - Waveforms resulting from capacitive coupling (at transmitter and receiver of aggressor and victim lines)**

### E.2.2.2 Inductive coupling

In the following, inductive coupling is modeled as an inductor in series with each signal, with some coupling factor K representing the extent to which the inductors' magnetic fields overlap. In effect these two inductors constitute a transformer, creating a stepped-down version of the aggressor signal on the victim line. The amplitude of the signal produced on the victim line is proportional to the rate of change in current (di/dt) on the aggressor line. Since the impedance of a transmission line is resistive, for points in the middle of a transmission line di/dt will be proportional to dV/dt. Because the crosstalk signal produced across the inductance in the victim line is in series with the transmission line, it has a different sign at each end of the inductor. Because the current in an inductor always opposes the magnetic field that produced it, the polarity of the crosstalk signal is reversed from the polarity of the di/dt on the aggressor line that produced it. As a result of these two facts, inductive crosstalk creates a pulse of forward crosstalk with polarity opposite to the edge on the aggressor, and a pulse of reverse crosstalk with the same polarity as the aggressor edge. Figure E.12 is a schematic of a model for inductive coupling. Figure E.13 shows waveforms resulting from inductive coupling at the sender and recipient component I/Os of the aggressor and victim lines.

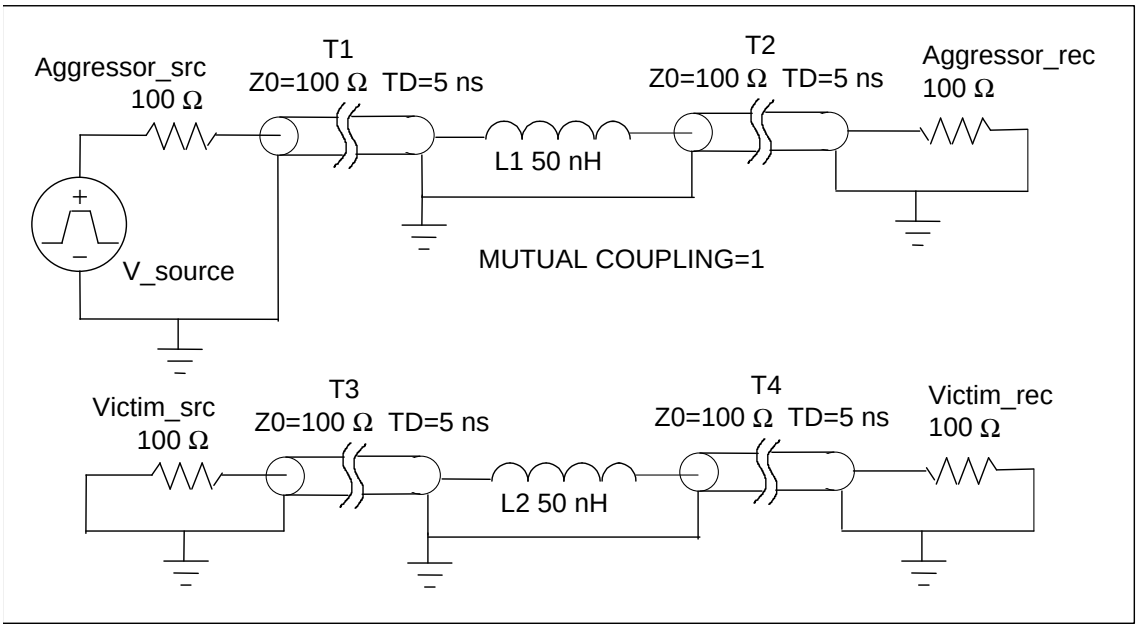



**Figure E.12 - Model of inductive coupling**

Note that the box in figure E.12, figure E.14, and figure E.18 between L1, L2 and K2 is a PSPICE element representing the inductive coupling between L1 and L2 having the coupling value listed in the figure.

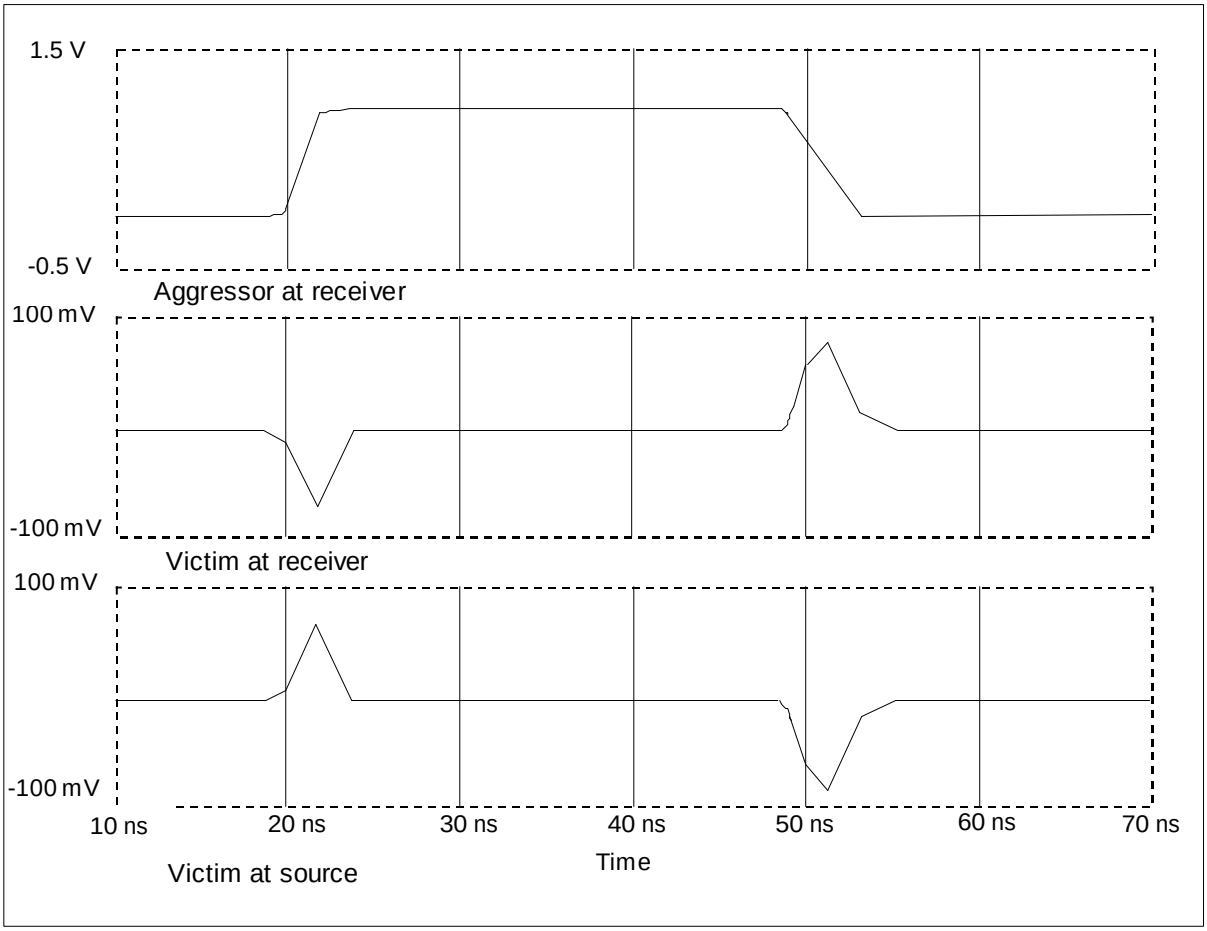



**Figure E.13 - Waveforms resulting from inductive coupling (at transmitter and receiver of aggressor and victim lines)**

### E.2.2.3 Mixed capacitive and inductive coupling

Most occurrences of electromagnetic coupling involve both capacitive and inductive coupling. In this case the forward and reverse crosstalk contributions of the capacitance and inductance add together. Because the forward inductive crosstalk and the forward capacitive crosstalk have opposite signs, they tend to cancel, while the reverse crosstalk from both effects have the same sign and add together. Depending on the ratio of inductive to capacitive coupling, the forward crosstalk may sum to zero when both effects are added together. Figure E.14 is a schematic of a model for mixed capacitive and inductive coupling. Figure E.15 shows waveforms resulting from mixed capacitive and inductive coupling at the sender and recipient component I/Os of the aggressor and victim lines.

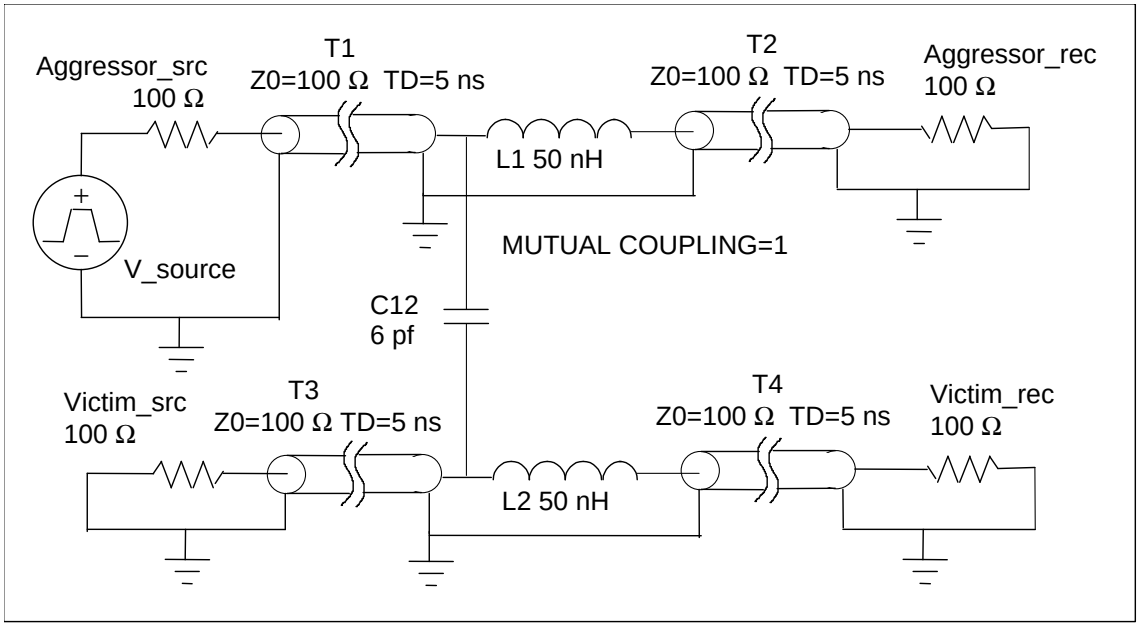



**Figure E.14 - Model of capacitive and inductive coupling**

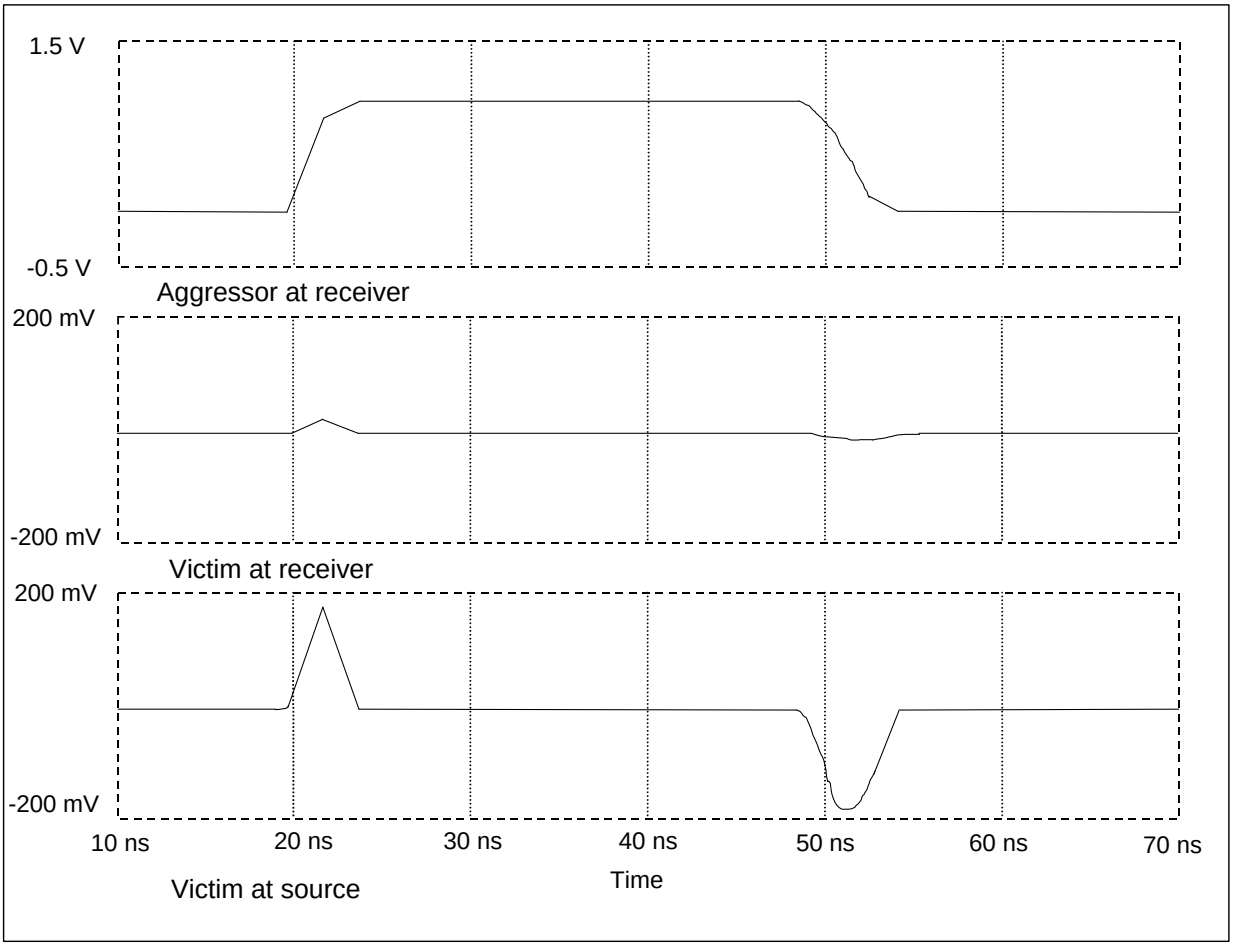



**Figure E.15 - Waveforms resulting from mixed capacitive and inductive coupling (at transmitter and receiver of aggressor and victim lines)**

### E.2.2.4 Crosstalk from distributed coupling

When transmission lines are placed parallel with and in close proximity to each other, as is the case for PCB traces, wires in a ribbon cable, etc., the coupling that occurs is continuous along the length of the transmission lines. To find the crosstalk waveforms at the source and recipient, divide the transmission lines into segments and treat each segment as an instance of capacitive and inductive coupling. Each segment produces forward and reverse crosstalk as the aggressor edge goes by. Sum the contributions from each of these segments, delaying their arrival at the ends according to the segment's position along the transmission line. This procedure shows that the forward crosstalk contributions all add together and arrive simultaneously with the aggressor edge, while the reverse crosstalk is spread out along the length of the transmission line and produces a long flat pulse travelling back toward the source. Figure E.16 shows a schematic model for a transmission line with three coupled conductors, connected as two signal wires and a ground return. The waveform at the source end of the victim line in figure E.17 shows that the reverse crosstalk pulse begins when the edge is driven onto the aggressor line and continues to be observed at the source until one system delay after the end of the edge is terminated at the recipient on the aggressor line. The waveform at the victim recipient's component I/O shows that the forward crosstalk arrives simultaneously with the edge on the aggressor line, or even slightly before, because the energy in the crosstalk pulse has been subtracted from the edge on the aggressor, reducing its rise time at the recipient.

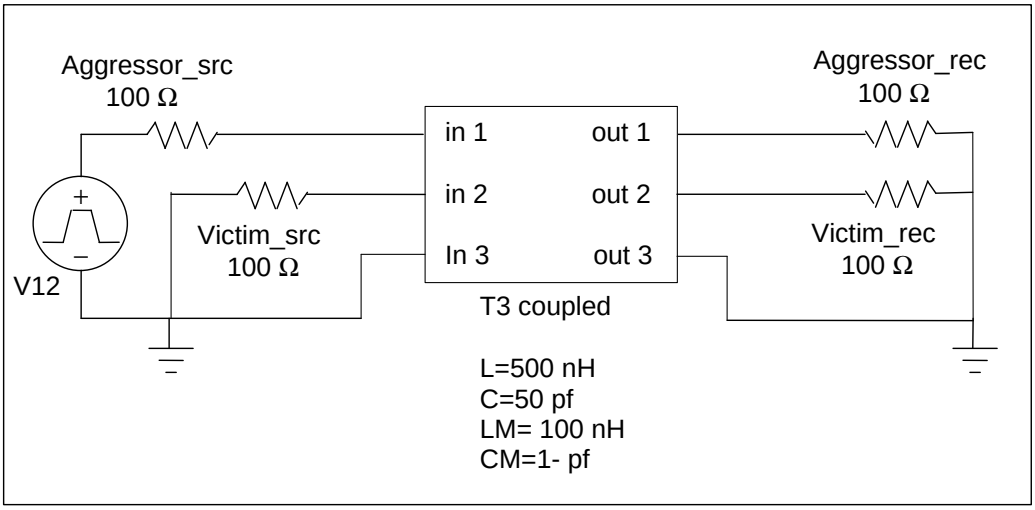



**Figure E.16 - Model of distributed coupling**

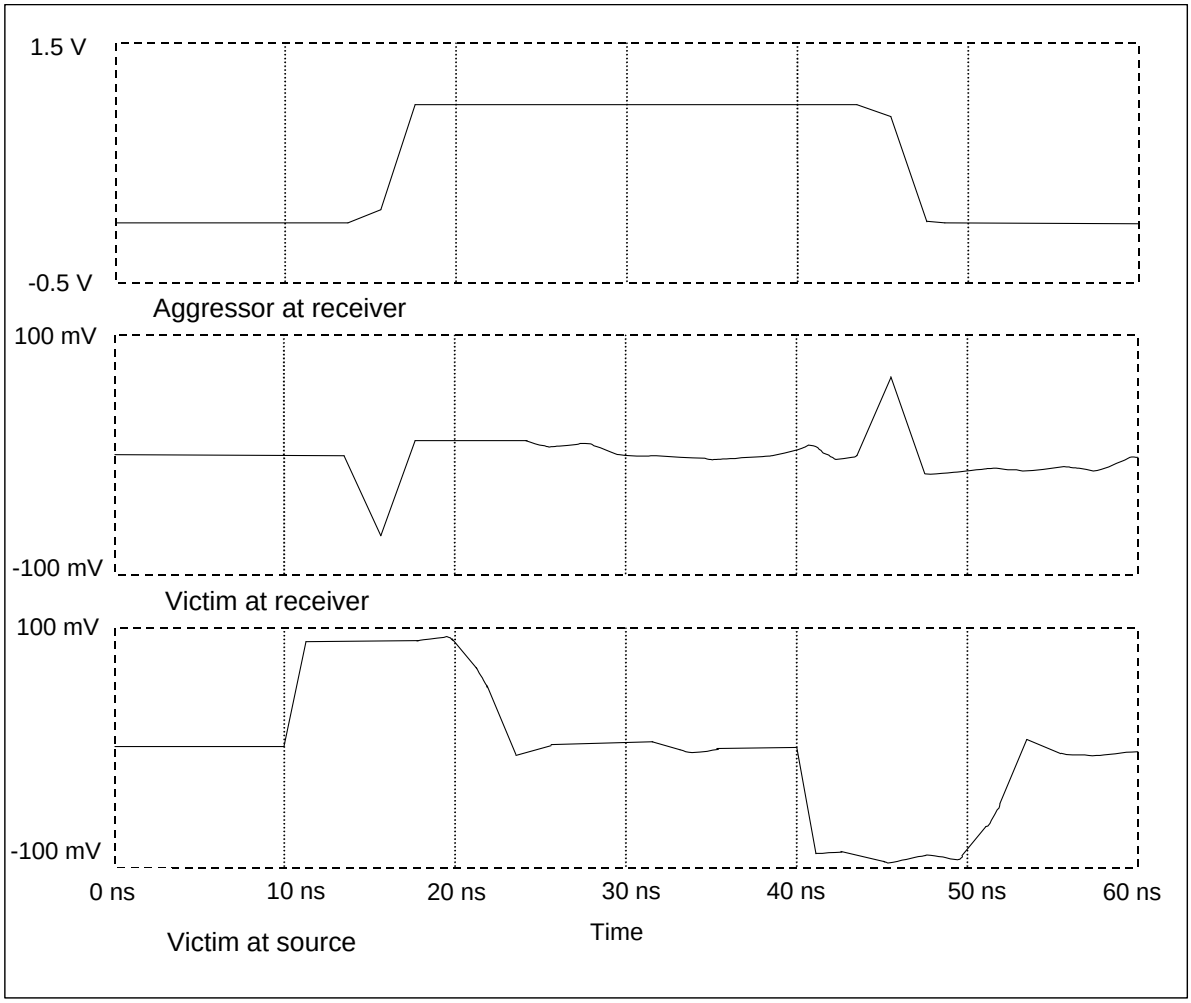



**Figure E.17 - Waveforms resulting from distributed coupling (at transmitter and receiver of aggressor and victim lines)**

The above simulation results shown in figures E.11, E.13, E.15, and E.17 are simplified by the assumption that all transmission lines are perfectly terminated at both ends. In actual systems only the sender end of the bus has a low-impedance termination to ground, and this termination is seldom perfect. The consequences of this help to explain some characteristics of crosstalk in a system:

1) Crosstalk is produced by both the initial and reflected edges on the aggressor lines. Forward crosstalk produced by the initial edge as it propagates from the sender to the recipient arrives at the same time as the edge that produced it. The edge on the aggressor signals reflects from the high impedance at the recipient input (or at the end of the cable) and returns back to the sender. Reverse crosstalk produced as this reflected edge propagates back to the sender is observed on the victim line at the recipient.
2) If reverse crosstalk from the initial edge is not perfectly terminated at the sender's component I/O it will be reflected (with reduced amplitude) back towards the recipient. The quality of the sender's component I/O termination depends on the instantaneous output impedance of drivers as they are switching, as well as the on resistance of the drivers in the high or low state once they have completed switching. Since the source impedance is made up of the driver output impedance in series with the termination resistors, the most accurate source termination is achieved by using drivers with low output impedance combined with high value series resistors, creating a total output impedance near 75 Ω.
3) Crosstalk is observed with doubled amplitude at the high-impedance endpoint of the system (at the host input during read operations and at the device end of the cable during write operations) due to the reflection. Since crosstalk occurs as a pulse rather than a step, the initial and reflected portions

of the pulse only sum at the endpoint while the pulse is reflecting, and not at other points along the bus.

4) Series termination resistors at the receiving end of the bus serve to attenuate the amplitude of crosstalk observed at the receiving component I/Os. Because the component I/O impedance is predominantly capacitive, its impedance decreases at high frequencies. At the frequency where the impedance of the component I/O equals the impedance of the series termination resistor, the crosstalk pulse amplitude observed at the IC input will be about half of the amplitude measured at the connector. The formula for determining this frequency is $F = 1 / (2 * \pi * R * C)$ where F is the frequency, R is the value of the series termination resistor, and C is the input capacitance of the recipient's component I/O. So when crosstalk levels are high enough to be a serious concern, the best place to make measurements of the crosstalk is at the component I/O or on the IC side of the termination resistor. In design of systems, this filtering effect is used to reduce a system's susceptibility to crosstalk by increasing the value of series termination resistors and placing them close to the connector to maximize the amount of capacitance on the IC side of the resistor.

In systems using the 80-conductor cable the largest contributors to crosstalk are the connector at the sender, and the PCB traces in systems with long traces or a large amount of coupling between traces. The connector at the receiving end of the system generates less crosstalk than the one at the sending end because the net current flow through the aggressor lines is less at the receiving end. This is because the load on the IC side of the recipient's connector is the PCB trace and a small capacitance inside the component I/O; only enough current flows through the connector to charge this total capacitance. At the sending end of the system, the instantaneous value of current through the connector is determined by the input impedance of the cable, and this amount of current flows for a length of time sufficient to charge the entire system including the cable and all attached devices up to the sender's $V_{oH}$ .

Crosstalk in the connectors is almost entirely inductive. It is produced in both directions from the connector but not necessarily in equal amplitudes. The highest amplitude crosstalk is generated by many switching lines coupling into a small number of victim lines. This lowers the effective source impedance of the crosstalk, making it approximate a voltage source. This voltage source is in series with the transmission line impedance on each side of the connector on the victim line. As a result, the crosstalk voltage is divided between the two directions proportional to the impedance seen in each direction. Figure E.18 shows the schematic of a model that demonstrates this. The PCB and cable on the victim line have been replaced with resistors to simplify the resulting waveforms. Figure E.19 shows the current through the inductor on the aggressor line and the crosstalk voltage produced on the victim line into the resistors representing the PCB and cable impedance. The waveforms indicate that the crosstalk voltage divides in the expected ratio. In this example the PCB receives $(50 / (82 + 50)) * 100\% = 37.9\%$ of the total voltage across the inductor, while the cable receives the remaining 62.1%. In an actual system, the crosstalk at the source is terminated by the driver impedance. The crosstalk measured at the recipient's component I/O on the victim line is double the value of the crosstalk pulse initially produced into the cable impedance.

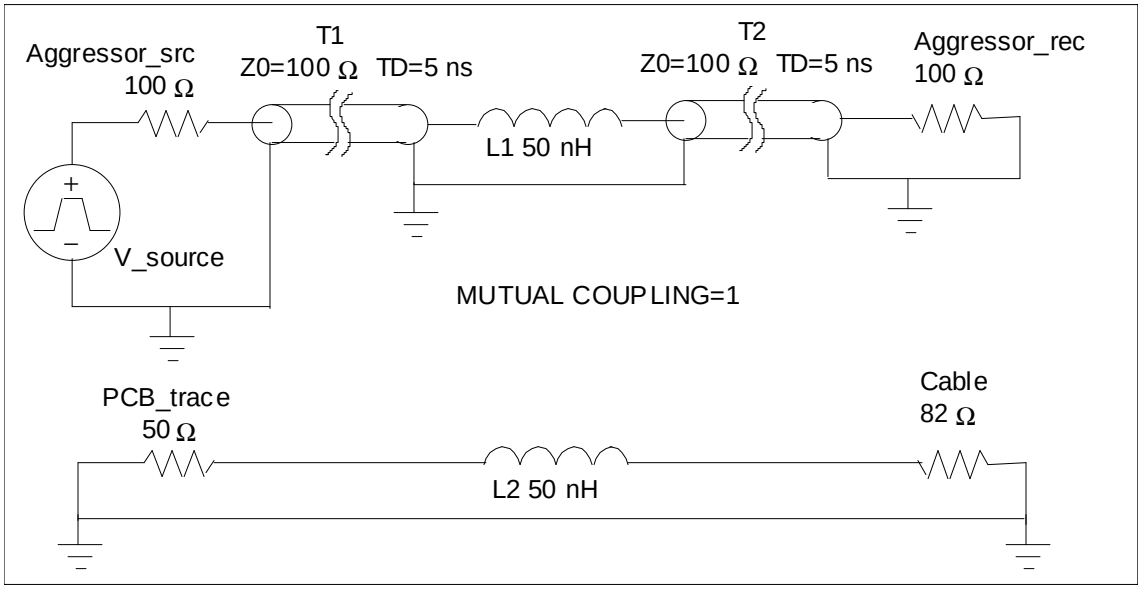



**Figure E.18 - Model of voltage divider for connector crosstalk formed by PCB and cable**

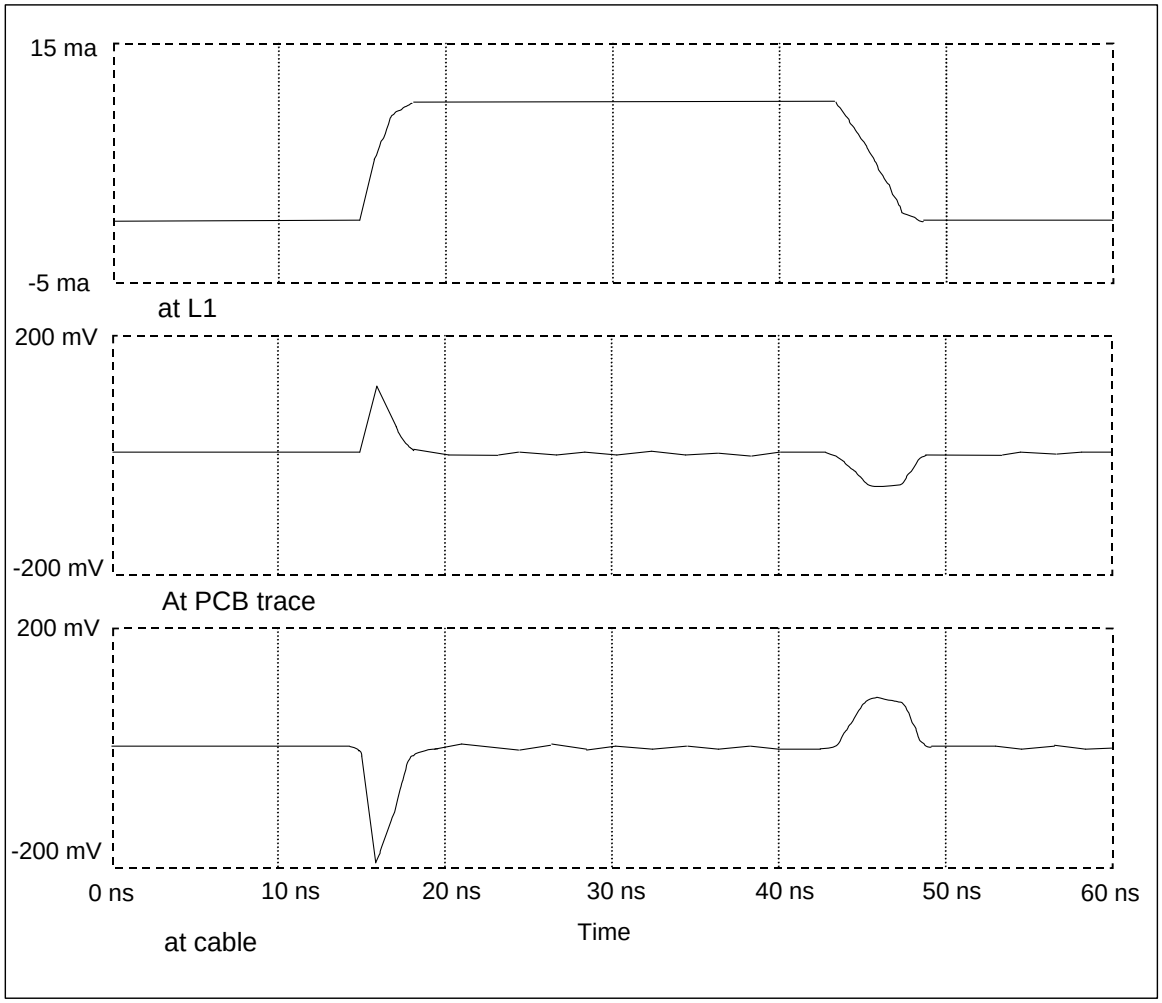



**Figure E.19 - Waveforms showing connector crosstalk dividing between PCB and cable**

For each edge on the bus four crosstalk pulses are created on non-switching victim lines due to the combined crosstalk in the PCB, connector, and cable:

1) Forward crosstalk from the initial edge has the same sign as the edge and is seen at the recipient as a pulse that arrives with the edge. The amplitude of the pulse is doubled at the recipient's component I/O, however because it occurs during the interval when the data is changing it may decrease the signal's setup or hold time but it presents a minor risk to data integrity overall.
2) Reverse crosstalk from the initial edge travels back towards the driver as a flat pulse with a width equal to the transition time of the driver. Based on the degree of mismatch between the driver's output impedance and the cable impedance, this pulse may be reflected back towards the recipient with reduced amplitude. Because it continues to arrive at the recipient well after the driver has completed switching, it creates a risk of incorrect data at the recipient in the middle of the cycle. However, this edge will seldom create a high enough amplitude at the recipient to cause a problem.
3) Forward crosstalk from the reflected edge arrives back at the driver simultaneously with the reflected edge on the aggressor lines. Depending on the impedance mismatch at the source, the edge will be reflected back towards the recipient with reduced amplitude and arrives in the middle of the cycle, however this edge will seldom create a high enough amplitude at the recipient to cause problems.
4) Reverse crosstalk from the reflected edge on the aggressor lines will be created travelling back toward the recipient and arrives there in the middle of the cycle. In host systems where the termination resistors are not placed next to the connector a larger portion of the crosstalk created in the connector will be reverse crosstalk on the cable side because of the divider formed by the 50 to 60 Ω PCB and the 82 Ω cable impedance. The pulse will be seen with doubled amplitude by the device at the end of the cable and presents a serious hazard to data integrity if its amplitude at the recipient's component I/O exceeds 800 mV.

#### E.2.2.5 Measuring crosstalk in a system

To measure the total crosstalk in a system set up a data pattern in which one line in the middle of the data bus is held low while all other lines are asserted simultaneously. Measure the low line at the recipient connector or component I/O. This measurement includes ground bounce at the sender IC discussed in E.2.3 as well as the contributions to crosstalk of the PCBs, connectors, and cables. Determining the exact sources of the different features of the crosstalk measured by this technique is difficult. An effective method to isolate the crosstalk produced into a victim line in a given portion of the system is to sever the line before and after the feature being tested. Terminate the isolated segment to ground at the breaks with resistors equivalent to the transmission line impedance that is normally seen at those points. Measuring the crosstalk voltage across the termination resistors will indicate the raw quantity of crosstalk into the victim line produced by that portion of the system, independent of reflections due to impedance mismatches and attenuation due to capacitance along the bus. Adjusting for impedance mismatches and delays will allow the crosstalk from that portion to be identified in the total crosstalk of the system, and adjusting the impedance changes through the system may allow the impact of that crosstalk to be minimized.

#### E.2.2.6 System design considerations to minimize crosstalk

Because all crosstalk throughout the system is proportional to edge rate, a major factor in controlling crosstalk is controlling the output slew rate of the drivers. Another major factor is the impedance match of sources to the cable including the value and placement of termination resistors. Source impedance matching is important to prevent reverse crosstalk from reflecting off the source and out to the recipient. Drivers, PCB layout, and termination resistors are selected to provide a good source termination for crosstalk and the reflected signal edge. Ideal termination at each connector is when the impedance seen looking back toward the source matches the cable impedance in the forward direction. For devices, this means that the sum of driver output impedance and termination resistance match the cable impedance (typically 80 to 85 Ω), minus five to ten percent to allow for attenuation due to the capacitive loading of other devices on the cable. Because the PCB traces on a device are short, they have little effect on the device's output impedance.

Due to other design constraints, many hosts PCB traces are so long that, for high-frequency crosstalk, the impedance at the host connector is determined by the PCB trace impedance and termination resistors (if they are located at the connector), rather than by the driver's output impedance. Because of this, there are two options for hosts with longer traces to ensure an ideal source termination:

1) Place the termination resistors near the sender's component I/O and use a PCB trace impedance that matches the source impedance of the sender's component I/O plus termination resistor. This

ideal impedance is slightly less than the cable impedance. In this case, trace impedance of 70 to 75 Ω with a large enough trace spacing to keep crosstalk (especially reverse crosstalk) between PCB traces to a minimum is ideal.

2) Place the termination resistors near the connector and select PCB trace impedance and termination resistance to sum to the cable impedance or slightly less. In this case, matching the sender's component I/O source impedance to the PCB trace impedance rather than the cable impedance is ideal, since that is the load that is driven.

Option 2 is desirable for backward compatibility with older systems using the 40-conductor cable because placing the resistor near the connector helps to damp the ringing that occurs with that cable. In addition, 50 to 60 Ω traces are easier to implement and produce less crosstalk than higher impedance traces making the second option a better choice in most cases.

In either case, matching the total output impedance to the cable impedance under all conditions of steady-state or switching is the best solution.

### E.2.3 Ground/Power Bounce

Supply bounce is a form of crosstalk that results from changes in current through power and ground pins of IC packages. For single-ended drivers, the return current for all signals flows through the power and ground leads, with the result that any voltage drop across these pins is imposed on all signals equally. Voltage drops across these pins occur due to both resistance and inductance whenever there is a net current flow into or out of the signal pins of the IC, though inductance has the greatest effect. In terms of the voltage seen at the recipient's component I/O, crosstalk due to supply bounce is indistinguishable from inductive crosstalk, with a sign opposite the polarity of the edge on the aggressor signal(s). See figure E.20 for a model of ground bounce in an IC package. See figure E.21 for waveforms resulting from ground bounce at the sender's and recipient's component I/O of the aggressor and victim signals.

In order to measure supply bounce in a functioning system, it is necessary to remove all other sources of crosstalk (especially reverse crosstalk from points later in the system). To remove the other sources of crosstalk, disconnect the component I/O pin on which the measurement is being taken from the PCB and measure the voltage at the component I/O while all other lines are switching. The initial and the reflected edges on the switching lines will produce supply bounce. Measurements with the victim line in a high state show power bounce and with the victim line in a low state show ground bounce. The ground inside the IC will bounce and produce crosstalk on a low victim line when many lines are switching from high to low and sinking current through the ground pins. The power inside the IC will bounce and produce crosstalk on a high victim line when many lines are switching from low to high, and drawing current through the power pins.

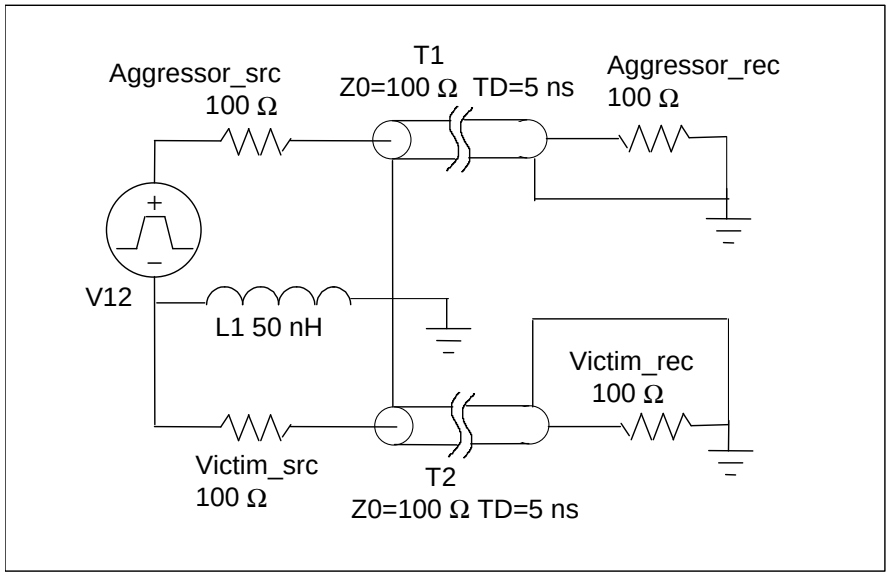



**Figure E.20 - Model of ground bounce in IC package**

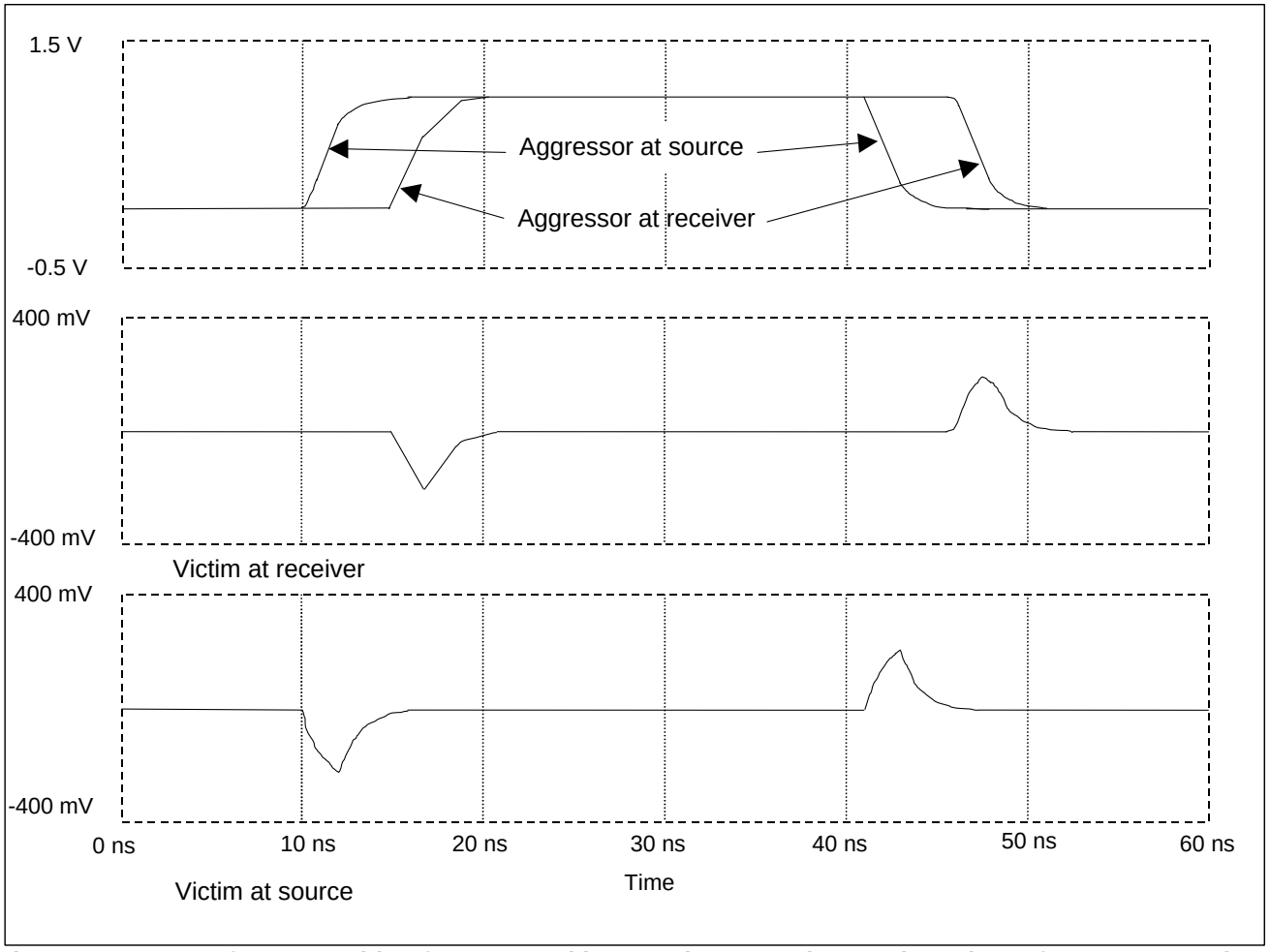



**Figure E.21 - Waveforms resulting from ground bounce (at transmitter and receiver of aggressor and victim lines)**

In order to reduce the susceptibility of Ultra DMA mode 5 to crosstalk, a tighter specification of input thresholds is defined. This Ultra DMA mode 5 requirement prevents strobing an incorrect data value due to

crosstalk with a peak amplitude less than 1.5 V positive from ground or negative from the minimum $V_{dd}$ of 3.3 V.

### E.2.4 Ringing and data settling time (DST) for the 40-conductor cable assembly

High amplitude ringing may occur for some data patterns in systems using the 40-conductor cable assembly. The sixteen data lines (DD(15:0)) in a 40-conductor cable assembly are adjacent to each other and have only one ground on each side of the data lines. There are only seven ground lines present in the entire cable assembly. This lack of ground return paths has three negative effects on data signal integrity:

1) Crosstalk between data lines is very high due to inductive coupling.
2) Conductors in the center of the set of data lines (e.g., DD 11) exhibit very high inductance because the distance from these signal lines to the current return path is large and the ground return path is shared with many other signal lines.
3) Conductors in the center of the set of data lines are shielded from ground by the other data lines around them. When these lines are switching in the same direction there is no potential difference and therefore no effective capacitance between lines.

This combination of factors results in the impedance of the conductors in the center of the set of data lines rising from 110 to 150 Ω (measured when a single line is asserted or negated) to an almost purely inductive 300 to 600 Ω when all lines are asserted or negated simultaneously in the same direction. Measured impedance varies with data pattern, edge rate, cable length, loading, and distance from chassis ground.

Unlike the 40-conductor cable, the 80-conductor cable has the additional 40 ground lines making all signals ground-signal-ground. This makes the 80-conductor cable impedance relatively constant with respect to pattern. Matching impedance and controlling PCB trace geometry as discussed in E.3.4 will result in well damped ringing and crosstalk in victim lines that remains below 800 mV.

In the following simplified model of the 40-conductor cable assembly with all data lines switching, a conductor in the center of the set of data lines is described as a pure inductor, forming a series RLC resonant circuit with the capacitance of the component I/O and PCB traces, and the combined resistance of the driver source impedance and source series termination resistor (see figure E.22). The voltage across C will ring sinusoidally in response to an input pulse at V_source, exponentially decaying over time towards a steady state value. The formula for determining the frequency of this ringing is $F = 1 / (2\pi * SQRT(LC))$ where F is the frequency, R is the value of the series termination resistor, and C is the input capacitance of the recipient's component I/O. The rate of decay is proportional to R/L. Figure E.23 shows the output of a simple RLC model with the waveforms as seen at the connectors of the sender and recipient.

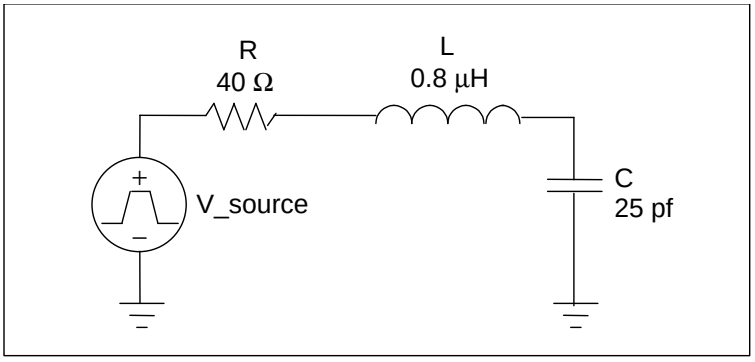



**Figure E.22 - Simple RLC model of 40-conductor cable with all data lines switching**

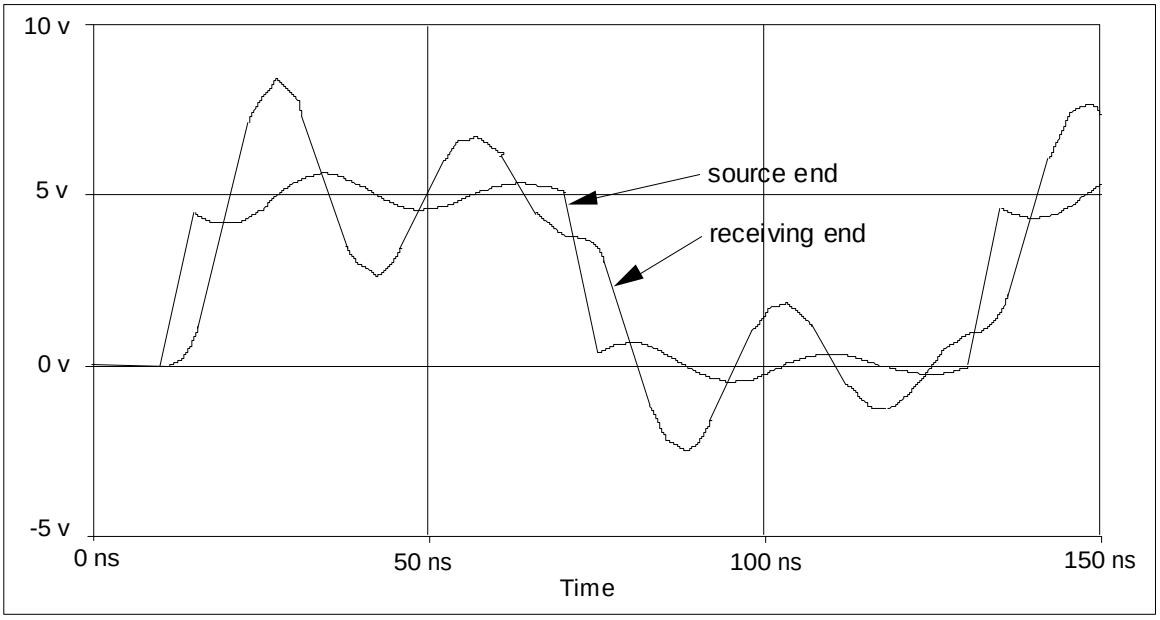



**Figure E.23 - Output of Simple RLC model: waveforms at source and receiving connectors**

DST is defined as the portion of cycle time required for ringing to decrease in amplitude until a signal reaches the threshold of 2.0 volts ($V_{iH}$) or 800 mV ($V_{iL}$). The worst-case situation for most systems occurs when all data lines are switching except for one line near the middle of the bus that is being held low (see figure E.24).

In this situation crosstalk creates a pulse on the signal line being held low that rings with a frequency and damping determined by the effective RLC parameters of the system. The DST value is the duration of time between the nominal beginning of the cycle (i.e., when the switching lines cross the 1.5 volt threshold) and the time when the ringing on the line drops below $V_{iL}$ for the last time as measured at the recipient's component I/O.

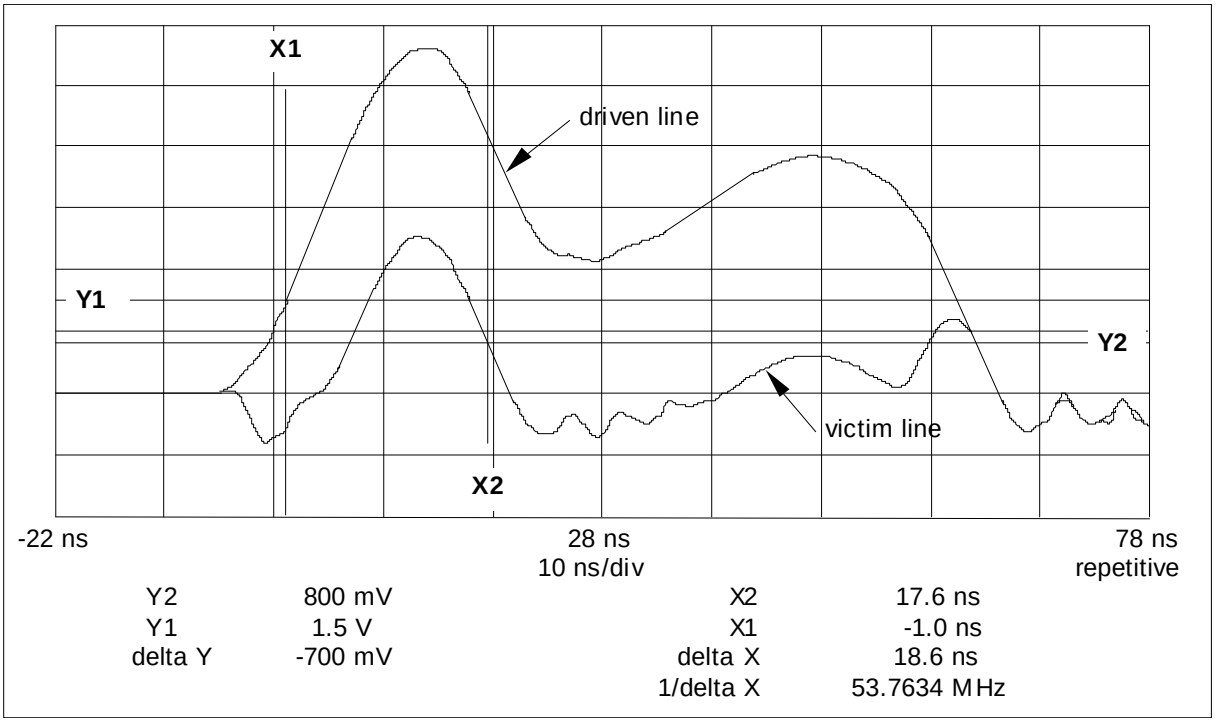



**Figure E.24 - DST measurement for a line held low while all others are switching high (ch1 on DD3 at rec., ch2 on DD11 at rec.)**

The same situation also occurs with reversed signal polarity (e.g., one line staying high while others are switching). Another case arises when all lines are switching simultaneously and the voltage on conductors in the center of the set of data lines rings back across the switching threshold (see figure E.25). This is normally only a problem in the high state as low side ringing is greatly reduced by the substrate diode clamp to ground that is inherent in CMOS logic.

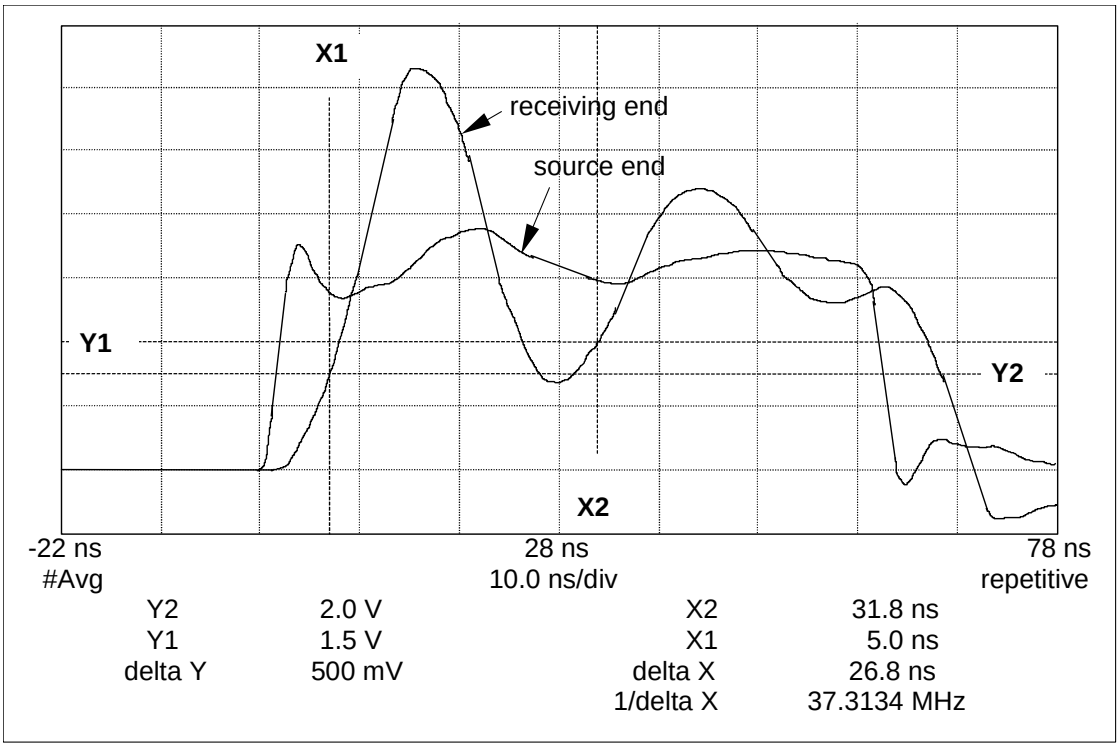



**Figure E.25 - DST measurement for all lines switching (ch1 at source, ch2 at rec.)**

As seen in figure E.25, the use of 3.3 volt signaling removes the high side voltage margin provided by the asymmetric threshold of the recipient input. Consequently it is important to use slew rate controlled drivers to control ringing.

### E.2.4.1 Controlling ringing on a 40-conductor cable assembly

An improved RLC model allows comparison between different termination schemes (see figures E.26 and E.27). These models include separate capacitors to represent trace and component I/O capacitance at the recipient's component I/O, as well as a clamping diode, representing the substrate diode in CMOS logic. Because this single-line simplified model does not include crosstalk between lines in the data bus, it is not used to predict DST for a particular design and combination of parameters. However, it does indicate the direction of changes in ringing frequency and damping in response to changes in system parameters.

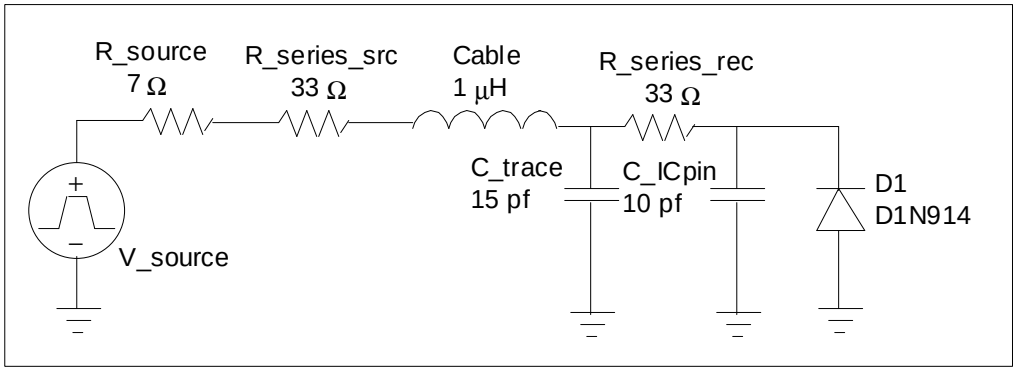



**Figure E.26 - Improved model of 40-conductor cable ringing with termination at IC**

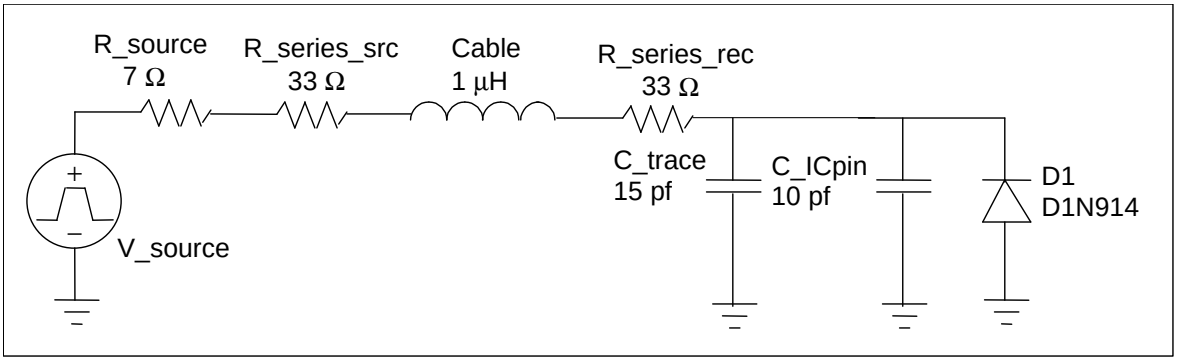



**Figure E.27 - Improved model of 40-conductor cable ringing with termination at connector**

Comparing the results (figure E.28) given by these models for recipient termination resistors located at the IC versus the connector shows that greater damping is provided when termination is near the connector.

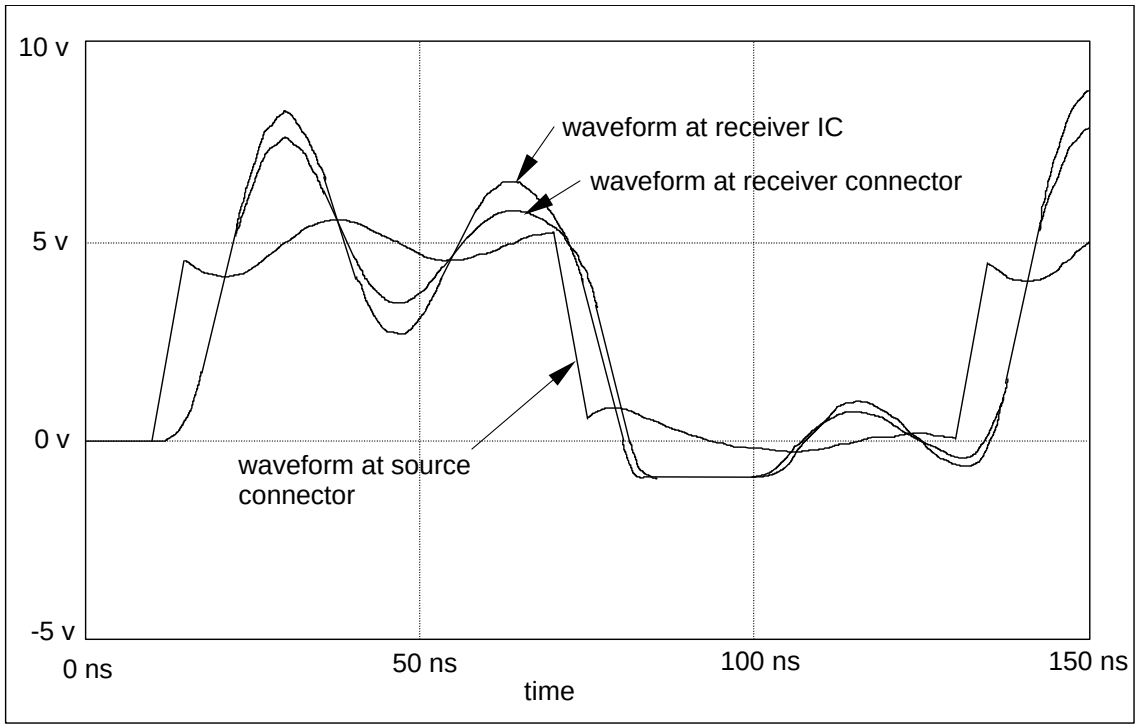



**Figure E.28 - Results of improved 40-conductor model with termination at IC vs. connector**

These simple models are used in a similar way to determine the effects of changing slew rate, termination resistor value, output impedance, PCB trace length, or the length of the cable.

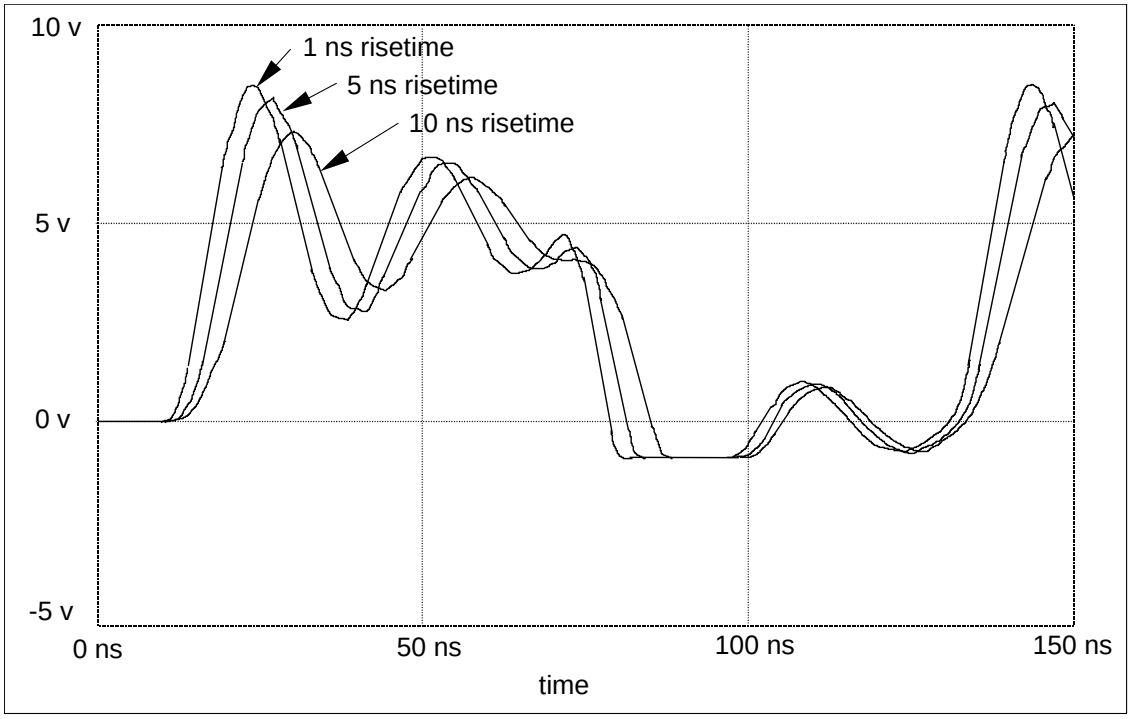



**Figure E.29 - Results of improved 40-conductor model with source rise time of 1,5,and 10ns**

As the results in figure E.29 show, increasing the rise time to above 5 ns results in a large decrease in the amplitude of the ringing. Drivers with control over the shape of rising and falling edges are used to reduce ringing even more.

Figures E.28 and E.29 show that, although the diode clamps the voltage at the recipient at one diode drop below ground, a ringback pulse appears at around 100 ns. This pulse occurs because the combined series resistance of the termination resistor and diode is much lower than the impedance of the LC circuit that is ringing. In addition the diode only clamps the voltage across part of the capacitance involved in the ringing. A higher-resistance clamping diode would be more effective at dissipating energy from the resonant circuit but would be less effective at clamping the input voltage.

#### E.2.4.2 STROBE lines on the 40-conductor cable

Although the data bus on the 40-conductor cable has such a high level of crosstalk that transmission line effects are barely perceptible, the STROBE lines on the 40-conductor cable have a more controlled impedance of about 115 Ω because they are in a ground-signal-ground configuration. Although the STROBE lines are well shielded against crosstalk from each other and from the data bus, some devices using drivers with fast edge rates and no source termination resistors have experienced problems with overshoot and ringback on the STROBE lines. Ringing will occur when a large impedance mismatch exists between the driver output impedance and the 115 Ω transmission line. If the ringback on a falling edge exceeds 800 mV, STROBE may cross the threshold multiple times and cause extra words to be clocked at the recipient. After these problems were experienced almost all device and host manufacturers began using series termination resistors on the STROBE lines at both the sender and the recipient.

With current component I/O technology and the requirement for series termination resistors, ringing on the STROBE lines is seldom a problem for current systems. However, it is important to keep in mind that these are high speed edge triggered signals, and the possibility of double crossing of input thresholds due to noise, ringing, or transmission line reflections still exists. Because of this it is important that all hosts and devices implement some amount of hysteresis on STROBE inputs in addition to glitch filtering by digital logic after the inputs.

## E.3 System Guidelines for Ultra DMA

This is a summary of recommendations for device, system, and chipset designers. These guidelines are not strict mandates, but are intended as tools for developing compatible, reliable, high-performance systems.

### E.3.1 System capacitance

All hosts and devices are required in the body of the standard to meet maximum values of capacitance as measured at the connector. These values are specified to be 25 pf at the host and 20 pf at the device. With typical interface IC and PCB manufacturing technology, this limits host trace length to four to six inches. It is recommended that capacitance be measured at 20 MHz as this is representative of typical ringing frequencies on an 18-inch 40-conductor cable assembly.

PCB traces up to 12 inches long may be used if the following conditions are met:

1) The host chipset uses 3.3 volt signaling,
2) The host chipset allows timing margin for the additional propagation delay in all delay-limited interlocks,
3) Termination resistors are chosen to minimize input and output skew and are placed near the connector,
4) Total capacitance of traces, additional components, and host component I/Os is held to the minimum possible, and
5) An 80-conductor cable is installed for operation at Ultra DMA modes 2 and higher.

In this case capacitance at the connector will exceed the maximum value specified. As a result of this, systems may not operate reliably with a 40-conductor cable assembly in any Ultra DMA mode above mode 1 (22.2 megabytes per second). Under these conditions it is advisable that a host not set mode 2 or above without insuring that an 80-conductor cable assembly is installed in the system.

### E.3.2 Pull-up and pull-down resistors

For hosts supporting Ultra DMA mode 5, the pull-up on IORDY is a 4.7 kΩ resistor to 3.3 V rather than a 1 kΩ resistor to 5 V. Other pull-up resistors on devices and hosts may be to 3.3 V or to 5 V. Pull-up and pull-down resistors should never have nominal values lower than the value specified by the standard.

Placement of pull-up and pull-down resistors on the source side of the series termination minimizes loss of DC margin due to pull-up/pull-down current through the series termination resistors.

### E.3.3 Cables and connectors

Exceeding a spacing of six inches between device connectors on an 80-conductor cable will cause increased skew when signaling to or from the device not at the end of the cable. As spacing between the devices decreases, the capacitance of the two devices (or the host and the device not at the end of the cable) act in parallel, resulting in decreased ringing frequency and increased DST.

In systems using a 40-conductor cable assembly, provide a continuous electrical connection from ground on the device chassis through the system chassis to the ground plane on the host PCB. Routing the cable in close contact with the chassis will reduce data settling time, as long as it is done without significantly increasing the cable length.

### E.3.4 Host PCB and IC design

As has been stated, matching the total output impedance of hosts and devices to the cable impedance is ideal to minimize reflections and reverse crosstalk due to the impedance mismatch between the PCB and cable. The impedance of the 80-conductor cable is specified to fall within the range of 70 to 90 Ω and is between 80 and 85 Ω for typical cables with solid wire and PVC insulation.

Keeping the ratio of PCB trace spacing to height above ground plane high helps to control crosstalk between traces.

Controlling PCB trace characteristics to minimize differences in propagation delay between STROBE and all DATA lines limits the skew. Factors that affect the delay are:

1) Trace length;
2) Additional capacitance due to stubs, routing on inner layers, pads, and external components such as pull-up resistors and clamping diodes; and
3) Additional inductance due to vias, series components such as termination resistors, and routing across a break in the ground plane, over areas with no ground plane, or at a larger height above the ground plane.

In systems using an 80-conductor cable to support Ultra DMA modes 3 and higher, series termination values shold be selected based on the impedance of the driver design during switching to meet the required output impedance into the cable.

Place series termination resistors as close as possible to the cable header or connector.

Choose series termination values to equalize input RC delays for the STROBE and DD(15:0) lines. For typical host IC implementations the same type of component I/O is used on all signals and therefore all termination resistors at both STROBE and DD(15:0) may have the same value.

Use sufficient ground and power pins on interface ICs to control supply bounce when many lines are switching at the same time.

**E.3.5 Sender and recipient component I/Os**

The 80-conductor cable assembly impedance is less than half that of the typical 40-conductor cable assembly impedance when multiple lines are switching at the same time. For some types of drivers this will result in more than double the current draw during switching and as a consequence the amplitude of ground bounce will also double.

As is required in this standard, design drivers to have a slew rate between 0.4 and 1.25 V/ns across the full range of loading conditions, process, and temperature.

Design component I/Os to produce output setup and hold times at the connector as specified in this standard across the full range of loading conditions, process, and temperature. Provide margin to allow for skew introduced between the IC and the connector. Design device PCB traces and component I/Os to present similar loading between STROBE and DD(15:0) at the connector to minimize additional skew added to signaling between other devices on the bus.

Use hysteresis on both DD(15:0) and STROBE inputs. Initial voltage steps on the bus are at undefined levels and may be near the thresholds, causing slow slew rates through the threshold that result in high sensitivity to noise if hysteresis is not used.

Test drivers as well as host and device output characteristics at the connector with the following loading conditions:

1) 0 pf to ground (open circuit, minimize test fixture capacitance)
2) 15 pf to ground
3) 40 pf to ground
4) 470 Ω to ground, switching low to high (simulates the 40-conductor cable with all lines switching)
5) 470 Ω to $V_{cc}$, switching high to low (simulates the 40-conductor cable with all lines switching)
6) 82 Ω to ground, switching low to high (simulates the initial edge into an 80-conductor cable)
7) 82 Ω to ground, switching high to low (simulates the initial edge into an 80-conductor cable)

All tests (except open circuit) are conducted with the intended series termination resistance in place. Output skew and slew rates are measured between the series termination and the load.

## E.4 Ultra DMA electrical characteristics

Along with the electrical characteristics defined for Ultra DMA modes 0 through 4, additional electrical characteristics have been added for Ultra DMA modes higher than 4.

### E.4.1 DC characteristics

The ATA interface was originally designed to use 5 V signaling. All pull-up resistor values are defined based on 5 V $V_{cc}$, and many hosts and devices still use 5 V $V_{oH}$ signaling levels. The advantage to using 5 V I/O cell technology is that the I/O cells will be 5 V tolerant. There are, however, many disadvantages to using a 5 V $V_{oH}$. One disadvantage is timing. The standard defines typical timings using a 1.5 V threshold. 5 V $V_{oH}$ is 2 V further from this threshold than 0 V $V_{oL}$. With identical rising and falling slew rates, the time for a signal to transition from 5 V to 1.5 V on a falling edge will be longer than the time for a signal to transition from 0 V to 1.5 V on a rising edge. For example, with an average slew rate of 500 mV/ns, a falling edge could take 4 ns longer to get from 5 V to 1.5 V than a rising edge to get from 0 V to 1.5 V. In fact, typical TTL thresholds are centered even lower than 1.5 V, at 1.3 V or 1.4 V, making the situation worse for timing to actual thresholds. Setup and hold time margins must account for both rising and falling edge skew. The skew with a 5 V $V_{oH}$ signaling level may be minimized by using asymmetric slew rates, but this poses its own difficulties and results in higher overall skew over process and voltage variations as does skew using 3.3 V signaling.

Another disadvantage to 5 V signaling is higher crosstalk. Given a 5 V transition and a 3.3 V transition having the same rise time, the 5 V transition will generate more crosstalk than the 3.3 V transition because the 5 V transition will result in higher di/dt and dv/dt to achieve the same rise time as the 3.3 V transition. Given a 3.3 V transition and 5 V transition of the same slew rate (dv/dt), the 5 V transition will still have higher crosstalk than the 3.3 V transition because the 5 V transition will be transitioning for a longer period of time.

A third disadvantage to 5 V $V_{oH}$ levels is overshoot. Given the high edge rates and low output impedance commonly observed on the ATA interface, a 5 V edge will nearly double at the receiver because the interface is series-terminated and the receiver is high impedance. In cases where edges are generated on DD(15:0) closer than they should be for any Ultra DMA mode (less than 15 ns apart), the ringing may be pumped even higher, in one observed case to nearly 11 V. Many 5 V tolerant parts are not designed to handle this level of overshoot.

#### E.4.1.1 $V_{iH}$ maximum

Since outputs supporting Ultra DMA modes higher than 4 are required to have 3.3 V $V_{oH}$ levels, it is still important for a designer to remember that many hosts and devices use 5 V $V_{oH}$ levels. Since a 5 V device may be connected to a 3.3 V host, that host and any other device attached to the same bus must be 5 V tolerant. Since a 3.3 V device may be connected to an existing 5 V host, all devices must be 5 V tolerant. To emphasize the requirement for 5 V tolerance, a $V_{iH}$ maximum of 5.5 V has been added to the standard. Also added was a note stating that ringing may generate AC voltages higher than the DC maximum $V_{iH}$.

#### E.4.1.2 $V_{DD3}$ (modes higher than 4 only)

Data setup and hold timings for Ultra DMA modes higher than 4 are more stringent than for lesser modes, and the starting $V_{oH}$ voltage is critical for low skew. Since $V_{DD3}$ of the sender and recipients typically do not track, the input threshold will not track with the $V_{oH}$ level. Without a defined supply voltage variation, it is difficult for an IC designer to determine if output and input I/Os will meet the required output and input skews required to meet system timing. For this reason the limits were defined for the 3.3 V supply. It should be noted that these limits are as measured at the interface IC. More careful control of the voltage may be required from the supply in order to meet these limits at the IC.

#### E.4.1.3 $V_{oH2}$ minimum and maximum (modes higher than 4 only)

The specification for the minimum and maximum $V_{oH2}$ levels requires that 3.3 V I/O cells be used. The minimum $V_{oH2}$ ensures that the I/O cell has a low enough on resistance to 3.3 V for proper termination with an 80-conductor cable. The maximum $V_{oH2}$ ensures that when the I/O cell is enabled and the bus is above 3.3 V, the I/O will pull the bus down to 3.3 V. This prohibits the use of a 5 V I/O cell and the use of I/O cell structures that have high impedance when enabled in the high state with an output voltage of more than 1 diode drop above the output supply voltage. It is important that the output both pull up and pull down to 3.3 V so that starting voltage is always at or near 3.3 V no matter what the initial condition on the bus. Starting at a voltage at or near 3.3 V for all transitions will help insure low skew for all signal transitions.

#### E.4.1.4 V+ and V- thresholds (modes higher than 4 only)

CRC errors may occur in some systems operating at Ultra DMA modes higher than 4 if the thresholds are too low and crosstalk occurs that exceeds the threshold. This may cause a zero to be strobed as a one. CRC errors due to excessive skew and setup or hold violations may also occur when the thresholds are low and negative going transitions beginning at beginning 5 V do not cross the threshold on the incident edge or its first reflection. Other transfer errors have been observed that resulted from thresholds that were set too high. A V+ minimum of 1.5 V was specified to ensure that with normal levels of crosstalk, bounce, and ringing at $V_{iL}$, the low-to-high threshold is not crossed. A V- maximum of 1.5 V was specified to ensure that with normal levels of crosstalk, bounce, and ringing at $V_{iH}$, the high-to-low threshold is not crossed. The minimum V- was set above the maximum $V_{iL}$ level of 0.8 V to be consistent with thresholds centered on 1.5 V and a maximum V+ of 2.0 V.

The average between the V+ and V- thresholds has also been specified with $V_{THRAVG}$. This value is used to ensure that the thresholds remain centered at 1.5 V. Shifting in the thresholds from center results in larger skew on rising verses falling inputs. Some shift is allowed to account for threshold shifts due to voltage, process, and temperature variation but less than would be allowed by the V+ and V- minimum and maximum specifications themselves. The specified $V_{THRAVG}$ value ensures that, with linear and symmetric rising and falling edges that cross 1.5 V at the same time once the first threshold is crossed, the other edge will be no more than 400 mV from its actual switching threshold.

#### E.4.1.5 Hysteresis ($V_{HYS}$) (modes higher than 4 only)

Hysteresis on STROBE and DD(15:0) inputs is required so that small amounts of noise on the signal do not cause input data capturing logic or the CRC value generation logic to double clock. Cases where the wrong CRC value is generated but the correct data is captured have been observed that resulted from double clocking of the CRC generator. Cases where the correct CRC value was generated but the wrong data was captured due to lack of hysteresis on the STROBE input have also been observed. Hysteresis on DD(15:0) decreases crosstalk and improves noise immunity.

### E.4.2 AC characteristics

High levels of ringing and crosstalk on the interface are due to the AC characteristics of the sender. While additional DC characteristics have been defined so that recipients are more tolerant of high crosstalk and ringing, requirements have been added to reduce generation of these by the sender for Ultra DMA modes higher than 4. This provides more margin, higher reliability, and better compatibility with older components that do not meet the Ultra DMA mode 5 and higher electrical specifications.

#### E.4.2.1 $S_{RISE2}$ and $S_{FALL2}$ (modes higher than 4 only)

Maximum slew rate is defined for Ultra DMA mode 4 with $S_{RISE}$ and $S_{FALL}$. However, it was determined that these maximums were not sufficient to reduce crosstalk to levels where victim signals maintain guaranteed low or high states. Simulations and data from many Ultra DMA mode 4 systems was used to determine the maximum slew rate for modes higher than 4, which is lower than for modes 0 through 4. The values for $S_{RISE2}$ and $S_{FALL2}$ were chosen to prevent the crosstalk from exceeding 800 mV at $V_{oL}$.

#### E.4.2.2 $V_{DSSO}$ and $V_{HSSO}$ (modes higher than 4 only)

Restricting the slew rate only is not sufficient to reduce the crosstalk maximum. Following good layout practices, like the guidelines in this document, and using I/O cells with sufficient power and ground pins for the maximum current and change in current that occurs during a transition is also important. $V_{DSSO}$ and $V_{HSSO}$ measure crosstalk from the sender's IC through the sender's connector during a simultaneously switching output (SSO) condition. This accounts for the majority of crosstalk in a system. Since the host typically requires longer trace lengths than devices, the $V_{HSSO}$ value is larger. A sender that meets this SSO maximum and the maximum slew rate should not produce excessive crosstalk at the recipient for a recipient that also meets its SSO maximum as a sender.

#### E.4.2.3 $C_{ratio}$ (modes higher than 4 only)

This is the ratio between the STROBE input capacitance and input capacitance of data signals. Sometimes different I/O cells are used for STROBE than for the bi-directional data lines, or additional loading is placed on some of the data lines. Under these conditions, the RC time constant of data input lines through the series termination could be very different than the time constant of the STROBE input. This could generate a large skew between data and STROBE. The $C_{ratio}$ value was defined to reduce the chance of high skew due to large loading variation and to encourage the use of the same I/O cell for STROBE and DATA inputs.

## E.5 Ultra DMA timing and protocol

### E.5.1 Ultra DMA timing assumptions

#### E.5.1.1 System delays and skews

Many of the system delays and skews used to determine timing specifications for Ultra DMA modes 0 through 4 were also used to determine mode 5 timing specifications. However, more stringent specifications on IC electrical characteristics are required to operate at the transfer rate of modes higher than 4. This resulted in improvements in the specifications. These improvements apply to any Ultra DMA mode at which an Ultra DMA mode 5 or higher capable host or device is operating, but are noted separately in the following for operation at modes higher than 4. Timing values are also included for some cases where a single device at the end of the cable provides additional improvement.

##### E.5.1.1.1 Source termination resistor delays

Min rising source transition delay = 0.34 ns (1.9 ns for modes higher than 4)
Min falling source transition delay = 0.23 ns (1.9 ns for modes higher than 4)
Max falling source transition delay = 2.61 ns (2.7 ns for modes higher than 4)

##### E.5.1.1.2 Recipient termination resistor delays

Max rising recipient transition delay = 0.12 ns
Max falling recipient transition delay = 0.12 ns

##### E.5.1.1.3 Transmission skews and delays

All skew values are the STROBE delay minus the data delay. The value for maximum negative skew is the minimum STROBE delay minus the maximum data delay for a worst-case system configuration. The value for maximum positive skew is the maximum STROBE delay minus the minimum data delay for a worst-case system configuration. The worst case system configurations were determined through simulation and include all possible system configurations that meet the requirements of the standard. Included in these values are skew due to variation in PCB trace length, PCB trace impedance, recipient component I/O capacitance, sender and recipient series termination, pattern, and common mode capacitance. Unless otherwise noted, timings are measured at 1.5 V.

Sender's component I/O to recipient's component I/O actual thresholds max negative skew = –5.99 ns (-4.34 ns for mode 5 and -3.6 ns for mode 6)

Sender's component I/O to recipient's connector max negative skew = –3.98 ns (-2.69 ns for mode 5, -2.29 ns for mode 6, -1.72 ns for mode 5 with single device only at the end of the cable, and -1.31 ns for mode 6 with single device only at the end of the cable)

Sender's component I/O to recipient's component I/O actual thresholds max positive skew = 5.38 ns (3.83 ns for mode 5, 3.06 ns for mode 6)

Sender's component I/O to recipient's connector max positive skew = 3.42 ns (1.83 ns for mode 5, 1.43 ns for mode 6, 1.11 ns for mode 5 with single device only at the end of the cable, and 0.71 ns for mode 6 with single device only at the end of the cable)

Sender's component I/O to recipient's component I/O maximum delay = 6.2 ns

For Ultra DMA modes 0, 1, and 2 using a 40-conductor cable, an additional -70, -36, and -22 ns are included in the two maximum negative skew values listed above to account for long data settle time due to crosstalk and ringing. The maximum positive skew values are not affected since the crosstalk and ringing on STROBE is not sufficient to increase its settle time.

**E.5.1.2 IC and PCB timings, delays, and skews**

It is recommended that the timing values shown in this clause be met but they are only an example of timing values that result in a system that meets all requirements for Ultra DMA specified in the standard. A system that does not meet one or more of the timing values below may be able to meet all timing requirements by producing other timing values more stringent than those shown below.

**E.5.1.2.1 Possible clocks for bus timing and their characteristics**

All frequencies are assumed to have 60 / 40 % asymmetry (worst case)

25 MHz (supports modes 0 and 1)
Typical Period = 40 ns
Clock variation = 1 %

30 MHz (supports modes 0, 1, and 2)
Typical Period = 33.3 ns
Clock variation = 1 %

33 MHz (supports modes 0, 1, and 2)
Typical Period = 30 ns
Clock variation = 1 %

50 MHz (supports modes 0, 1, 2, and 3)
Typical Period = 20 ns
Clock variation = 3.5 %

66 MHz (supports modes 0, 1, 2, 3, and 4)
Typical Period = 15 ns
Clock variation = 3.5 %

100 MHz (supports mode 0, 1, 2, 3, 4, and 5)
Typical Period = 10 ns
Clock variation = 4.0 %

133 Mhz (supports mode 0,1, 2, 3, 4, 5, and 6)
Typical Period = 7.5 ns
Clock variation - 3.0 %

**E.5.1.2.2 PCB traces**

The PCB trace skew value is based on 2.5 ns/ft propagation delay, 10 inch maximum host trace length, 2.5 inch maximum device trace length, and data trace lengths being +/- 0.5 inch STROBE trace lengths.

Max PCB trace skew = 0.1 ns
Max PCB trace delay = 2.1 ns

**E.5.1.2.3 IC inputs**

The input delay value includes values for bond wire, buffer, routing, and logic component delay between the input to the IC and the flip-flop that first latches the data. Input delay is measured from 1.5 V and includes the delay between 1.5 V and the input's threshold.

The value for input skew is either positive or negative depending on the direction of the STROBE and data transitions. This value is the difference in STROBE signal delay from the input switching threshold to the internal flip-flop that first latches data and data delay from the input switching threshold to the same flip-flop. The routing component of skew that accounts for about 30 % of the value listed here is systematic (i.e., always the same polarity in a system implementation) and could be either positive or negative.

Min input slew rate for testing = 0.4 V/ns
Max input delay = 5.5 ns (4.0 ns for modes higher than 4)
Max input skew = 2.45 ns (1.35 ns for modes higher than 4)
Max input skew from 1.5 V to actual thresholds with linear 0.4 V/ns input = 1.75 ns (1.0 ns for modes higher than 4)

**E.5.1.2.4 IC outputs**

Output delay is from the internal active clock edge that generates an output transition until the time that the transition crosses 1.5 V at the associated component I/O of the IC.

Max output disable delay is from the internal enable negation of an I/O output until the time that the signal is released at the component I/O.

Single component I/O output skew is the difference in delay of rising and falling edges on a single output. This single component I/O skew does include skew due to noise that may be present on the signal in a functional system. It may be positive or negative depending on the direction of the STROBE and data transitions.

Output skew is the difference in the output delay of the active STROBE and the output delay of any data transition that occurs within cycle time before or after the STROBE transition. This timing is met under all expected loading conditions and starting voltages. This timing is the combination of:

- single component I/O output skew,
- skew due to output routing differences between all data and STROBE signals,
- skew due to process, temperature, and voltage variation between all data and STROBE signals at the moments when transitions are generated,
- skew due to clock routing to all data and STROBE logic that generates output transitions, and
- skew due to supply bounce differences that may occur between the transitions being compared.

As with the single component I/O output skew, this skew may be positive or negative depending on the direction of the STROBE and data transitions. Some of the components of this skew (i.e. differences in routing) may be systematic but could be either positive or negative so are included in derivations using either.

Max output delay = 14 ns
Max output disable delay = 10 ns
Max single component I/O output skew = 2.5 ns (2.33 ns for mode 5, 1.54 ns for mode 6)
Max output skew = 5.4 ns (3.85 ns for mode 5, 2.28 ns for mode 6)

Max output skew to support modes 0 and 1 with a 25 MHz clock = 5.0 ns

Max output skew to support modes 0, 1, and 2 with a 50 MHz clock = 5.2 ns
Max output skew to support mode 4 with a 30 or 33 MHz clock = 2.8 ns

Noise skew = 0.45 ns (0.33 ns for mode 5, 0.29 ns for mode 6). Noise skew is part of the output skews above. It is also listed here so that the noise skew contribution can be removed from timings defined at the sender IC. However, this is intended to be used only for simulations that will not include high frequency noise coupled into the traces and cable. Noise skews for Ultra DMA modes 4 and higher are based on peak-to-peak noise of 0.18 V and 0.13 V respectively and a minimum slew rate of 0.4 V/ns except for mode 6 where a minimum 0.45 V/ns is assumed.

Up to 3 ns of additional output delay may be needed for data compared to STROBE in cases that use 30 and 33 MHz clocks to support Ultra DMA modes 0, 1, and 2. With these clocks, the data is held by a half cycle, and a minimum half cycle is not sufficient to meet the output hold time given the output skews listed above. An additional delay on data would insure that the required hold time is met even with a short half clock cycle. Alternatively, improvements in output skew beyond those listed above could also allow the output hold time to be met with a short half clock cycle.

#### E.5.1.2.5 IC flip-flops

The setup and hold times listed here are intended to represent only the flip-flops inside an IC that latch data. Timing is assumed from the inputs of the flip-flop.

Min flip-flop setup time = 0.5 ns (0.2 ns for modes higher than 4)
Min flip-flop hold time = 0.5 ns (0.2 ns for modes higher than 4)

### E.5.2 Ultra DMA timing parameters

System timings for all Ultra DMA modes are measured at the connector of the sender or receiver to which the parameter applies. Internally the IC accounts for input and output delays and skews associated with all signals getting from the connector to the internal flip-flop of the IC and from the flip-flop of the IC to the connector.

Timings as listed in the body of the specification were derived using the formulas listed below and the timing assumptions give above. All applicable clocks were evaluated for each timing parameter and the worst-case value was used in the body of the standard. It is recommended that the system designer re-derive all timings based on the specific characteristics of the internal clock, IC, and PCB that are to be used to confirm that timing requirements are met by that implementation.

#### E.5.2.1 Typical average two-cycle time ($t_{2CYCTYP}$)

This is the typical sustained average time of STROBE for the given transfer rate from rising edge to rising edge or falling edge to falling edge measured at the recipient's connector.

#### E.5.2.2 Cycle time ($t_{CYC}$)

This is the time allowed for STROBE from rising edge to falling edge or falling edge to rising edge measured at the recipient's connector. This timing accounts for STROBE and internal clock variation. The formula for the minimum value is:

+ (Number of clock cycles to meet minimum typical cycle time with a minimum cycle time due to clock variation) ∗ (clock cycle time)
- Max single component I/O output skew

$t_{CYC}$ should be measured at the recipient connector at the end of the cable. Measurement of this parameter at the sender connector is obscured by reflections on the bus.

### E.5.2.3 Two-cycle time ($t_{2CYC}$)

This is the time for STROBE for the given transfer rate from rising edge to rising edge or falling edge to falling edge measured at the recipient's connector. Since this timing is measured from falling edge to falling edge or rising edge to rising edge of STROBE, asymmetry in rise and fall time has no affect on the timing. Clock variation is the only significant contributor to $t2_{CYC}$ variation. The formula for the minimum values is:

+ (2 ∗ (Number of clock cycles to meet minimum typical cycle time with a minimum cycle time due to clock variation percent) ∗ (clock cycle time))

### E.5.2.4 Data setup time ($t_{DS}$)

This is the data setup time at the recipient. Since timings are measured at the connector and not at the component I/O, consider the effect of the termination resistors and traces when generating this number. Depending on the direction of the data signal and STROBE transitions, the skew between the two changes in both the positive and negative directions. A longer data signal delay will reduce the setup time, and a longer STROBE delay will increase the setup time.

In order to meet the input skews given above, minimize the number of buffers or amount of logic between the incoming signals and the input latch or flip-flop. This may require the data input buffers to be routed directly to the input latch with no delay elements and the STROBE signal to be routed directly from its input buffer to the input latch clock with no delay elements.

The internal latch or flip-flop has a non-zero setup and hold time. $t_{DS}$ is sufficient to insure that the setup time of the flip-flop is met. The minimum setup required at the threshold of the component I/O is:

+ Max input skew
+ Min flip-flop setup time

The formula for the value at the recipient's component I/O based on the timings given in E.4 is:

+ (Number of clock cycles to meet typical cycle time with a minimum cycle time due to clock variation) ∗ (clock cycle time)
– (Number of clock cycles used to hold data with a minimum cycle time due to clock variation or with a minimum cycle symmetry if a half cycle is used) ∗ (clock cycle time)
– Max output skew
+ Sender's component I/O to recipient's component I/O actual thresholds max negative skew

In order to meet both setup and hold times over process, temperature, and voltage, clock edges rather than gate delays are used to generate the hold time. The assumption is made that one 50 or 66.7 MHz clock cycle or half of a 33 MHz or slower clock cycle has been used to hold data within the sender IC.

After it is shown that the sender is producing a setup time that meets the requirement of the recipient, the specification for setup time at the recipient connector produced by the sender is determined as follows. The $t_{DS}$ values in the specification were based on the results of the following formula using all possible clocks for the modes they support. The $t_{DS}$ value for mode 5 was defined for a single device only located at the end of the cable in order for best determination of system margin during validation. A value for two devices attached to the cable is determined with the timings given above.

+ (Number of clock cycles to meet typical cycle time with a minimum cycle time due to clock variation) ∗ (clock cycle time)
– (Number of clock cycles used to hold data with a minimum cycle time due to clock variation or with a minimum cycle symmetry if a half cycle is used) ∗ (clock cycle time)
– Max output skew
+ Sender's component I/O to recipient connector max negative skew

### E.5.2.5 Data hold time ($t_{DH}$)

This is the data hold time at the recipient. This time is sufficient to insure that the hold time of the internal flip-flop is met. The longest STROBE delay and shortest data delay is the worst case for hold time. The analysis is similar to the one for $t_{DS}$ above. The minimum hold required at the component I/O at its threshold is:

+ Maximum input skew
+ Minimum flip-flop hold time

The formula for the value at the recipient's component I/O based on the timings given above is:

+ (Number of clock cycles used to hold data with a minimum cycle time due to clock variation or with a minimum cycle symmetry if a half cycle is used) ∗ (clock cycle time)
– Max output skew
– Sender's component I/O to recipient's component I/O actual thresholds max positive skew

After it is shown that the sender is producing hold time that meets the requirement of the recipient, the specification for hold time at the recipient connector produced by the sender is determined as follows. The $t_{DH}$ values in the specification were based on the results of the following formula using all possible clocks for the modes they support. The $t_{DH}$ value for mode 5 was defined for a single device only located at the end of the cable in order for best determination of system margin during validation. A value for two devices attached to the cable is determined with the timings given above.

– (Number of clock cycles used to hold data with a minimum cycle time due to clock variation or with a minimum cycle symmetry if a half cycle is used) ∗ (clock cycle time)
– Max output skew
– Sender's component I/O to recipient connector max positive skew

### E.5.2.6 Data valid setup time ($t_{DVS}$)

This is the data valid setup time measured at the sender's connector. This timing is measured using a test load with no cable or recipient. This is the timing that, if met by the sender, will insure that the data setup time is met at the recipient. It is important that this timing be met using capacitive loads from 15 to 40 pf to ensure reliable operation for any system configuration that meets specification.

In the case of Ultra DMA modes 0, 1, and 2, long data settle times occur due to crosstalk in the cable and on the PCB, and the ringing frequency of the system. For modes above 2, there is little or no margin for ringing on the cable. For these modes, the 80-conductor cable assembly that reduces the crosstalk between signals is required so that crosstalk and ringing are reduced to a level that does not cross the input switching thresholds during data setup or hold times. Modes 3 and 4 timing requirements were derived to be met with the same input and output timing characteristics as a system supporting Ultra DMA mode 2. Since the formulas presented for $t_{DS}$ show that sufficient setup time is produced with the given system timings, using those same timings in the formula below will produce $t_{DVS}$ values that also represent sufficient timing for the system. An achievable value for $t_{DVS}$ is calculated as follows:

+ (Number of clock cycles to meet minimum typical cycle time at the minimum cycle time due to clock variation) ∗ (clock cycle time)
– (Number of clock cycles used to hold data at the minimum cycle time due to clock variation or at the minimum cycle symmetry if a half cycle is used) ∗ (clock cycle time)
– Max output skew
– Max PCB trace skew
– Max falling source transition delay
+ Min rising source transition delay

### E.5.2.7 Data hold time ($t_{DVH}$)

This is the data valid hold time measured at the sender's connector. This timing is measured using a test load with no cable or recipient. This is the timing that, if met by the sender, will insure that data hold time at the recipient is met. It is important that this timing be met using capacitive loads from 15 to 40 pf to insure reliable operation for any system configuration that meets specification.

Since the formulas presented for $t_{DH}$ show that sufficient hold time is produced with the given system timings, using those same timings in the formula below will produce $t_{DVS}$ values that also represent sufficient timing for the system. An achievable value for $t_{DVH}$ is calculated as follows:

+ (Number of clock cycles used to hold data at the minimum cycle time due to clock variation or at the minimum cycle symmetry if a half cycle is used) ∗ (clock cycle time)
– Max output skew
– Max PCB trace skew
– Max falling source transition delay
+ Min rising source transition delay

### E.5.2.8 CRC word setup time ($t_{CS}$) (modes higher than 4 only)

For Ultra DMA modes 0 through 4 the value for $t_{CS}$ is the same as the value for $t_{DS}$. The formula and details used to determine $t_{CS}$ are identical to the ones used to determine $t_{DS}$. In order to determine the value for $t_{CS}$ for modes higher than 4, Ultra DMA mode 4 output, system, and input skew values are used rather than the mode 5 or 6 values. This is because DMACK- is used to strobe the CRC word rather than HSTROBE. Host system designers sometimes pay very close attention to the skew between HSTROBE and DD(15:0). The secondary function of DMACK- is as a clock for the CRC word. On the device, flip-flops to capture data will be carefully placed to minimize skew between incoming data and HSTROBE. Rather than require the device to add the additional flip-flops needed to capture the CRC word with respect to the DMACK- signal, the value for $t_{DS}$ for Ultra DMA mode 4 is specified.

### E.5.2.9 CRC word hold time ($t_{CH}$) (modes higher than 4 only)

For Ultra DMA modes 0 through 4 the value for $t_{CH}$ is the same as the value for $t_{DH}$. The formula and details used to determine $t_{CH}$ are identical to the ones used to determine $t_{DH}$. In order to determine the value for $t_{CH}$ for modes higher than 4, Ultra DMA mode 4 output, system, and input skew values are used rather than the mode 5 or 6 values. This is because DMACK- is used to strobe the CRC word rather than HSTROBE. Host system designers sometimes pay very close attention to the skew between HSTROBE and DD(15:0). The secondary function of DMACK- is as a clock for the CRC word. On the device, flip-flops to capture data will be carefully placed to minimize skew between incoming data and HSTROBE. Rather than require the device to add the additional flip-flops needed to capture the CRC word with respect to the DMACK- signal, the value for $t_{DH}$ for Ultra DMA mode 4 is specified.

### E.5.2.10 CRC word valid setup time ($t_{CVS}$) (modes higher than 4 only)

For Ultra DMA modes 0 through 4 the value for $t_{CVS}$ is the same as the value for $t_{DVS}$. The formula and details used to determine $t_{CVS}$ are identical to the ones used to determine $t_{DVS}$. In order to determine the value for $t_{CVS}$ for modes higher than 4, Ultra DMA mode 4 output, system, and input skew values are used rather than the mode 5 or 6 values for reasons given in the description for $t_{CS}$. If specified to be the same value for mode 5 as that specified for mode 4, $t_{CVS}$ for mode 5 would be less than the time of one 100 MHz clock cycle. Though a single system clock cycle was sufficient for $t_{CVS}$ for mode 4, consideration of all output skews demonstrates that the mode 4 setup time is not met with a single 100 MHz clock cycle. The value for modes 5 and 6 $t_{CVS}$ was set to 10 ns in an attempt to force host IC designers to hold the CRC value for two cycles in modes 5 and 6 rather than mistakenly assuming one would do.

### E.5.2.11 CRC word valid hold time ($t_{CVH}$) (modes higher than 4 only)

For Ultra DMA modes 0 through 4 the value for $t_{CVH}$ is the same as the value for $t_{DVH}$. The formula and details used to determine $t_{CVH}$ are identical to the ones used to determine $t_{DVH}$. In order to determine the value for $t_{CVH}$ for modes higher than 4, Ultra DMA mode 4 output, system, and input skew values are used rather than the mode 5 values for reasons given in E.5.2.9. If specified to be the same value for mode 5 as that specified for mode 4, $t_{CVH}$ for mode 5 would be less than the time of one 100 MHz clock cycle. Though a single system clock cycle was sufficient for $t_{CVH}$ for mode 4, consideration of all output skews demonstrates that the mode 4 setup time is not met with a single 100 MHz clock cycle. The value for mode 5 and 6 $t_{CVH}$ was set to 10 ns in an attempt to force host IC designers to hold the CRC value for two cycles in modes 5 and 6 rather than mistakenly assuming one would do.

### E.5.2.12 First DSTROBE time ($t_{ZFS}$) (modes higher than 4 only)

The protocol for every Ultra DMA mode relies on the fact that IORDY has a pull-up resistor at the host. It is true, however, that if the device chooses to generate the first high-to-low transition by switching the driver from released to negated (as is allowed by the protocol for Ultra DMA modes less than 5) that transition could have a longer delay than other high-to-low transitions. This is especially true in the case where there is an I/O buffer with a 3.3 V $V_{oH}$ level and a bus that is pulled up to 5 V. On subsequent high-to-low transitions, the transition would start at or near 3.3 V instead of at or near 5 V, which would result in a shorter fall time and better delay matching with the rising transition through the input thresholds.

In all modes less than mode 5, there is sufficient timing margin to use 5 V I/O cells. Many device and host implementations for Ultra DMA mode 4 use 5 V $V_{oH}$ outputs. In the case of an output with a 5 V $V_{oH}$, the released-to-negated transition may have about the same delay as an asserted-to-negated transition. With a 5 V output I/O cell having reasonable skew and a typical hold time of 15 ns, the first high-to-low STROBE transition could be generated by going from released to negated while still meeting the hold time minimum. The setup and hold timings for modes higher than 4 require much lower skews between rising and falling edges than previous modes. It is therefore required that in those modes, the STROBE first falling edge start at or near 3.3 V and is generated with an output that is settled.

The timing $t_{ZFS}$ minimum for modes higher than 4 requires the device to assert DSTROBE a specified time before the first negation. When enabled high, I/O cells used for modes higher than 4 are required to pull the bus down to a value at or near 3.3 V if the bus is higher than 3.3 V when enabled. Starting a negation from being asserted at or near 3.3 V will provide better symmetry between negating and asserting transitions than when starting at or near 5 V. I/O cells may also have some internal transistors and reference settle times after being enabled. If the first negation is generated by switching from released to negated, the transition will not be as well controlled as one generated after the output has been asserted long enough to settle. For modes higher than 4, DSTROBE must be asserted for the specified time before the first high-to-low transition in order for that transition to start at or near 3.3 V and for that transition to have the same edge rate control and delay as any other high-to-low transition. It should be noted that when DSTROBE is first asserted, it must not produce any glitch that would pull the bus down below the host's V- threshold. In that case, the host would incorrectly detect a DSTROBE transition.

In order to force the device to enable DSTROBE asserted before generating the first DSTROBE negation, the $t_{ZFS}$ timing parameter was added.

### E.5.2.13 Data enabled to the first DSTROBE edge time ($t_{DZFS}$) (modes higher than 4 only)

As described above, DSTROBE must be asserted for a sufficient time before first negation to provide good timing and slew rate for that negation. DD(15:0) must also be driven for a sufficient time before the first STROBE transition is generated for DD(15:0) to be settled long enough to meet the setup time and to generate good edges after the required hold time. If DD(15:0) are enabled for the typical setup time before the first STROBE transition, the setup time will not be met. As with DSTROBE, this is because a transition generated from released to negated or asserted will have a longer delay than a transition generated from negated to asserted or asserted to negated. The $t_{DZFS}$ timing for modes higher than 4 assumes a typical 10 ns hold time for DD(15:0) and, since this time is referenced to the STROBE transition, it assumes the same delay for DD(15:0) from enabled to first transition as $t_{ZFS}$ is for DSTROBE. For most cases, it is sufficient for

DD(15:0) to settle in time to meet the required setup time. However, if the released to asserted or released to negated time is longer than about 10 ns, a longer enabled to first STROBE time is required in order to meet $t_{DVS}$.

### E.5.2.14 First DSTROBE time ($t_{FS}$)

This is the time for the device to first negate DSTROBE to clock the first word of data for a data-in burst after the device has detected that the host has negated STOP and asserted HDMARDY-. This parameter is measured from the when both STOP is negated and HDMARDY- is asserted at the device connector until the first negation of DSTROBE at the device connector.

Synchronization may be achieved with two flip-flops. After synchronization is achieved, data is driven on to DD(15:0) and internal clock cycles counted to meet the minimum setup time before generating the first DSTROBE transition. In order for an IC based on a 25, 30, or 33 MHz clock to meet $t_{FS}$, DD(15:0) must be driven no later than 2.5 clock cycles after the control signal transitions. This could be achieved by synchronizing with both edges of the system clock or by using only one edge to synchronize and then driving data onto DD(15:0) on the next inactive edge of the clock after the signals are detected at the output of the second synchronization flip-flop. With a 50 MHz clock, the first word of data must be driven out no later than three cycles after the control transitions and with a 66 MHz clock, it may be four cycles. The formula for the maximum $t_{FS}$ timing is as follows:

+ Max falling recipient transition delay
+ Max PCB trace delay
+ Max input delay
+ Min flip-flop setup time
+ The time for two, three, or four clock cycles at the maximum period due to frequency variation to synchronize the control signals and start the data transfer cycle. For 25, 30, and 33 MHz systems, the data would be driven out one half cycle after the incoming signal is synchronized since data is held one half cycle when using these clock frequencies and therefore sent on a half cycle.
+ The time for as many cycles as required to meet the $t_{DVS}$ minimum timing for the first word of data at the maximum period due to frequency variation.
+ Max output buffer delay
+ Max PCB trace delay
+ Max falling source transition delay

### E.5.2.15 Limited interlock time ($t_{LI}$)

The time is for limited interlock from sender to recipient or recipient to sender. This is the interlock time in the Ultra DMA protocol that has a specified maximum. The value of $t_{LI}$ must be large enough to give a recipient of the signal enough time to respond to an input signal from the sender of the signal. The derivation of $t_{LI}$ is similar to that of $t_{FS}$ since both involve the recipient of the signal responding to the control signal of the sender of the signal. As with $t_{FS}$, the number of internal clock cycles that an IC may require before responding is dependent on the frequency of the clock being used. For a 25 or 30 MHz clock, the maximum time to respond is three cycles, for 33 MHz clock it is four, for a 50 MHz clock it is five, and for a 66 MHz clock it is seven cycles maximum for modes 0 through 2. Modes 3, 4, 5, and 6 require a faster response time. For a 30 or 33MHz clock it is two cycles, for a 50 MHz clock it is three cycles and for a 66 MHz clock it is four clock cycles maximum. The formula for the values of $t_{LI}$ is as follows:

+ Max falling recipient transition delay or max rising recipient transition delay
+ Max PCB trace delay
+ Max input delay
+ Min flip-flop setup time
+ The time for two, three, four, five, or seven clock periods (depending on clock used and modes supported) at the maximum period due to frequency variation to synchronize the signals to the internal clock and respond appropriately.
+ Max output buffer delay
+ Max PCB trace delay

+ Max falling source transition delay

### E.5.2.16 Limited interlock time with minimum ($t_{MLI}$)

This time is for the minimum limited interlock from sender to recipient. This timing insures that the respective control signals are in their proper state before DMACK- is negated. It is important that STROBE and the control signals are in their proper states because all signals revert to their non-Ultra DMA definitions at the negation of DMACK-. If the signals are not in their proper state, the selected device or another device may incorrectly interpret a STROBE signal. For all control signals to be in their proper state and detectable at the device before DMACK- is negated, $t_{MLI}$ must exceed the sum of the following:

+ Sender's component I/O to recipient's component I/O maximum delay
+ Max input delay
+ Min flip-flop setup time

The value calculated by the formula above for $t_{MLI}$ for all modes is under 14 ns. The specified value for this timing allows for additional margin.

### E.5.2.17 Unlimited interlock time ($t_{UI}$)

This interlock timing is measured from an action by a device to a reaction by the host. In order to allow the host to indefinitely delay the start of a read or write transfer, this value has no maximum. The reason for this parameter is to ensure that one event occurs before another, for this reason the minimum is set to zero. In practice the host will take some non-zero positive time to respond to the incoming signal from the device.

### E.5.2.18 Maximum driver release time ($t_{AZ}$)

This is the maximum time that an output driver has to make the transition from being asserted or negated to being released. During data bus direction turn around, the driver of DD(15:0) is required to release these signal lines. For the beginning of a read burst, the host releases DD(15:0) before or on the same internal clock cycle that it asserts DMACK-. For the end of a read burst, the device releases DD(15:0) before or on the same clock cycle that it negates DMARQ. If the same clock is used, the maximum delay is calculated using the following formula:

+ Max output skew
– Min falling source transition delay

The value calculated by the formula above for $t_{AZ}$ for all modes is under 6 ns. The specified value for this timing allows for additional margin.

### E.5.2.19 Minimum delay time ($t_{ZAH}$)

This is the minimum time that the host waits after the negation of DMARQ at the termination of a data-in transfer to begin driving data onto DD(15:0) for the purpose of transferring the CRC word to the device. In this case the device is allowed to continue driving DD(15:0) for a maximum of $t_{AZ}$ after the DMARQ negation. The host is required to wait $t_{ZAH}$ after the DMARQ negation to drive the data. Skew on the cable is the major factor to consider here and a longer data delay than DMARQ delay (i.e., max negative skew) is the worst case. For modes using a 40-conductor cable, the component of maximum negative skew associated with data settle time as described above should not be included since DD(15:0) is being released for this timing. To avoid bus contention, this value is calculated using the following formula:

+ Max specified $t_{AZ}$
- Sender's component I/O to recipient's component I/O actual thresholds max negative skew

The value calculated by the formula above for $t_{ZAH}$ is under 17 ns in all cases. The specified value for this timing allows for additional margin.

### E.5.2.20 Minimum driver assert/negate time ($t_{ZAD}$)

This is the minimum time after STOP is negated or HDMARDY- is asserted (whichever comes later) that a device drives DD(15:0) at the initiation of a data-in burst. This is when DD(15:0) are changed from host driving or released to device driving.

The STOP negation and HDMARDY- assertion are required by the standard to meet $t_{ENV}$ timing that is a minimum of 20 ns from the point where the host releases DD(15:0). No additional delay is necessary based on the $t_{ZAH}$ evaluation that is applicable to the conditions of this timing. The device waits for STOP to be negated and HDMARDY- to be asserted and then may start driving DD(15:0).

The use of STOP negated and HDMARDY– asserted guarantees that a system failure will not occur leaving the host in a Multiword DMA mode and the device in an Ultra DMA mode. STOP is the same signal line as DIOW-, and HDMARDY- is the same signal line as DIOR-. The Multiword DMA protocol never allows assertion of both DIOW- and DIOR- at the same time. The negation of STOP and assertion of HDMARDY- is equivalent to both DIOW- and DIOR- being asserted. Since the device requires both signals to be in this state before driving DD(15:0), it insures that the host is in an Ultra DMA mode and not a Multiword DMA mode and has released DD(15:0).

Even though $t_{ZAD}$ has a 0 ns minimum for all modes, in practice, most devices will take two flip-flop delays to synchronize the incoming STOP and HDMARDY- transitions making the $t_{ZAD}$ time dependant on the clock frequency used by the device. Since DD(15:0) are driven long enough before the first STROBE to meet the setup time requirement, this synchronization time has been taken into account in the $t_{FS}$ derivation above.

### E.5.2.21 Envelope time ($t_{ENV}$)

This time is from when the host asserts DMACK- until it negates STOP and asserts HDMARDY- at the beginning of a data-in burst, and the time from when the host asserts DMACK- until it negates STOP at the beginning of a data-out burst. Since $t_{ENV}$ only applies to outputs from the host, the timings are synchronous with the host clock. Based on an argument similar to the one for $t_{MLI}$ in E.5.2.16, the minimum for $t_{ENV}$ is 20 ns. This insures that all control signals at all the devices are in their proper (non-Ultra DMA mode) states before DMACK- is asserted and are sensed as changing only after DMACK- has been asserted. The 20 ns accounts for cable and gate skew between DMACK- and the control signals on device inputs. Since $t_{ENV}$ involves synchronous events only and an increase in $t_{ENV}$ reduces the performance of the specification, a maximum is specified.

Enough internal clock cycles are used between the assertion of DMACK- and the other control signals to insure $t_{ENV}$ minimum is met. For a 25, 30, or 33 MHz clock this is a single cycle, for 50 or 66 MHz clocks this is two cycles. The following formula is used to verify that the minimum $t_{ENV}$ value of 20 ns is met by any particular system implementation:

+ (One or two host clock cycles (depending on frequency used) at the minimum period due to frequency variation to delay control signals inside the IC) ∗ (clock cycle time)
- Max output skew
- PCB trace skew
- Max falling source transition delay
+ Min falling source transition delay

The minimum is achieved by using the number of clock cycles specified above for each possible frequency. Based on the number of clock cycles needed to meet the minimum, reasonable maximums for $t_{ENV}$ are determined. Rather than limiting the possible cycles to generate $t_{ENV}$, the following assumption was made: for a 25 or 30 MHz clock a single cycle is used; for a 33 or 50 MHz clock a maximum of two cycles is used; and, for a 66 MHz clock a maximum of three clock cycles is used. Using these numbers of cycles, the formula to determine the maximum $t_{ENV}$ is as follows:

+ (One, two, or three cycles (depending on frequency used) at the maximum period due to frequency variation to delay control signals inside the IC) ∗ (clock cycle time)
+ Max output skew

+ PCB trace skew
+ Max falling source transition delay
− Min falling source transition delay

It may be possible that fewer or more clock cycles are used with some frequencies given reduced output skew. If the AC timing characteristics described above are just met, the following number of clock cycles for the internal IC delay to meet $t_{ENV}$ minimum and maximum values are used.

1) with 25 MHz, delay is one cycle
2) with 30 MHz, delay is one cycle
3) with 33 MHz, delay is one or two cycles
4) with 50 MHz, delay is two cycles
5) with 66 MHz, delay is two or three cycles (only two for modes above 2)
6) with 100 or 133 MHz, delay is three cycles

#### E.5.2.22 STROBE to DMARDY- time ($t_{SR}$)

If DMARDY- is negated before this maximum time after a STROBE edge, then the recipient will not receive more than one additional STROBE (i.e., one more word of valid data). This timing is applicable only to modes 0, 1, and 2 because the transfer rate of modes 3 and higher is too high to insure that only one additional STROBE will be sent after DMARDY- is negated.

Though there is no known implementation of the following method, this timing could be met by the recipient through the synchronization of the outgoing DMARDY- negation and the incoming STROBE signal from the sender. Design complexity would be added with little advantage. For this reason $t_{SR}$ was removed from the timing table. The asynchronous negation of DMARDY- with respect to the incoming STROBE is the preferred implementation. In this implementation, the negation of DMARDY- for pauses would be controlled by the state of the FIFO. Once a near-full condition occurs, DMARDY- could be negated. There is no advantage toward FIFO size in trying to meet $t_{SR}$ since synchronizing the outgoing DMARDY- signal with the incoming STROBE requires an additional STROBE to occur after a FIFO near-full condition is detected before the DMARDY- can be negated. If the asynchronous method is selected as recommended, then the recipient will always be ready for the maximum number of words allowed after it negates DMARDY-.

#### E.5.2.23 DMARDY- to final STROBE time ($t_{RFS}$)

This is the maximum time after DMARDY- is negated after which the sender will not transmit any more STROBE edges (i.e., no additional valid data words). This timing gives the sender time to detect the negation of DMARDY- and respond by not sending any more STROBES. The $t_{RFS}$ time may affect the number of words transferred.

Since $t_{RFS}$ involves a response to a request for a pause, the sender needs to stop sending data as soon as practical. An example of an input synchronization method is to use two flip-flops where the first is clocked on the active edge of the internal clock and the second on the unused (inactive) edge of the clock. The action to stop the STROBE signal would be taken on the next active clock edge (i.e., if there had been a STROBE scheduled for that edge it would not be sent). In this example a half cycle of the clock gives adequate time to avoid metastability while synchronizing the signal. The following timing diagram shows one possible case:

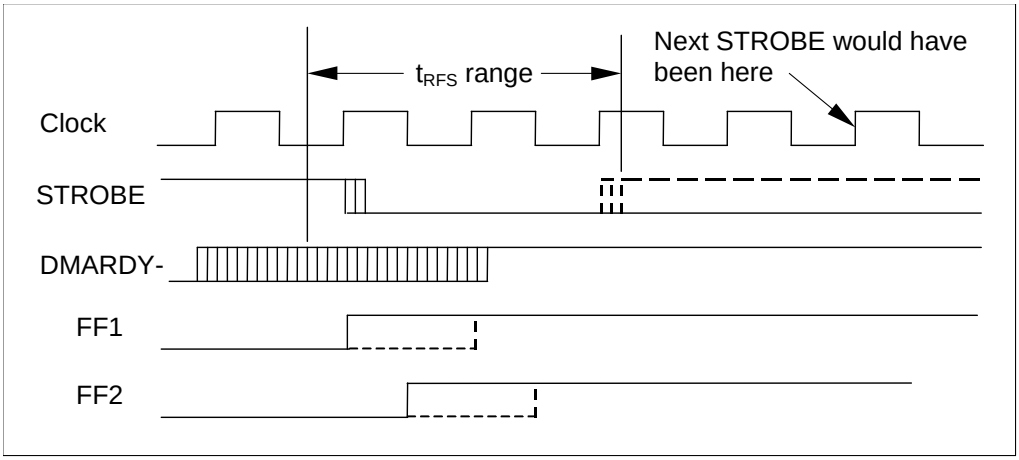



**Figure E.30 - DMARDY- to final STROBE $t_{RFS}$ synchronization**

Figure E.30 shows the range of possible STROBE to DMARDY- transition relationships and the possible synchronization flip-flop responses. When a 66 MHz or higher clock frequency is used, two clock periods may be used to synchronize the data as long as no STROBE edge is sent on the subsequent clock edges until the transfer is resumed.

The $t_{RFS}$ time may be the longest when the DMARDY- transition occurs before an internal clock cycle, but, due to skews and missed setup time, the transition is not clocked into the first flip-flop until the next clock (the dotted line transition on FF1 and later on FF2). When this happens one clock cycle before a STROBE transition is generated (as shown by the left $t_{RFS}$ range marker near the middle of the DMARDY- transition range in the diagram above), the next STROBE transition will occur (as shown in dotted lines). For all other cases, the $t_{RFS}$ time will be shorter. The maximum $t_{RFS}$ is calculated using the following formula:

+ Max rising recipient transition delay
+ Max PCB trace delay
+ Max input delay
+ Min flip-flop setup time
+ (One or two clock cycles at the maximum system clock period due to frequency variation for synchronization) ∗ (clock cycle time)
+ Max output delay
+ Max PCB trace delay
+ Max falling source transition delay

### E.5.2.24 DMARDY- to pause time ($t_{RP}$)

This is the minimum time after DMARDY- is negated after which the recipient may assert STOP or negate DMARQ-. After this time the recipient will not receive any more STROBE edges (i.e., no additional valid data words). STROBE edges may arrive at the recipient until this time. Since this time parameter applies to the recipient only as the recipient waits for STROBEs, the parameter is measured at the recipient connector. Because of this, the output delay of DMARDY- from inside the IC to the connector and the input delay of a STROBE edge from the connector to the associated internal IC flip-flop are considered.

There are two ways to determine the $t_{RP}$ minimum. One method is to consider how long it will take from the negation of DMARDY- at the recipient for the sender to see the negation and become paused. This would involve synchronizing DMARDY- as it is done for $t_{RFS}$, and then taking one more system clock cycle to change the state of the state machine to a paused state. Using this method, the minimum time is calculated using the following formula:

+ Sender's component I/O to recipient's component I/O maximum delay
+ Max input delay
+ Min flip-flop setup time

+ (Two or three clock cycles (depending on clock used) at the maximum period due to clock frequency variation) ∗ (clock cycle time)

A second method to calculate this value is to consider how long it might take for the last STROBE to be detected after negating DMARDY-, and make sure $t_{RP}$ is long enough so that the internal assertion of STOP occurs after the last STROBE has latched the last word of data. This method is applied in the following formula:

+ Sender's component I/O to recipient's component I/O maximum delay
+ Maximum $t_{RFS}$ for mode
+ Sender's component I/O to recipient's component I/O maximum delay
+ Max input delay
+ Min flip-flop setup time

Using both of the above, it may be shown that $t_{RP}$ is met given the $t_{RFS}$ requirement and is sufficient to receive the last STROBE for all modes with all clock frequencies. All of the values are measured at the connector, and the time to wait internal to the IC will be longer than the value of $t_{RP}$. For higher frequency clocks, the internal delay may need to be more than one clock cycle longer than the value of $t_{RP}$ in order to account for total output and input delays.

#### E.5.2.25 Maximum IORDY release time ($t_{IORDYZ}$)

This is the maximum time allowed for the device to release IORDY:DDMARDY-:DSTROBE at the end of a burst. The $t_{IORDYZ}$ time allows IORDY to be asserted immediately after DMACK- is asserted. DMACK- being asserted may be used to enable the IORDY output. As soon as the DMACK- is negated, the component I/O cell will be released. For this implementation, the following formula determines the maximum $t_{IORDYZ}$:

+ Max falling recipient transition delay
+ Max PCB trace delay
+ Max in delay (in this case to enable IORDY)
+ Max output disable delay
+ Max trace delay

#### E.5.2.26 Minimum IORDY assert time ($t_{ZIORDY}$)

This is the minimum time allowed for the device to assert IORDY:DDMARDY-:DSTROBE when the host asserts DMACK- at the beginning of a burst.

When STOP is negated and HDMARDY- is asserted, it is important that the IORDY:DDMARDY-:DSTROBE signal be electrically high (DSTROBE asserted or DDMARDY- negated). This could be achieved by the device driving the IORDY:DDMARDY-:DSTROBE signal, but it also occurs when this signal is released by the device because of the pull-up at the host required by the standard. Since the correct state of IORDY:DDMARDY-:DSTROBE occurs when it is released, no maximum $t_{ZIORDY}$ is required. As with some other timings having no maximum defined, the state of this signal will eventually be changed as governed by other timing parameters.

For Ultra DMA, DDMARDY-:DSTROBE is only driven during a data burst. At the initiation of a data-in burst, the device may wait until the time to generate the first DSTROBE and enable DSTROBE in a negated state. The device may wait $t_{ZIORDY}$ then assert DSTROBE and, for the first data transfer, the device would negate DSTROBE. In both cases the host sees a negation for the first DSTROBE. The first STROBE of a burst is never a low-to-high transition. At the initiation of a data-out burst, the device waits until ready before asserting DDMARDY-. If the device does not use this implementation, it waits $t_{ZIORDY}$ then negates DDMARDY- (i.e., drives it electrically high). Then, to signal that the device is ready to receive data, the device may negate DDMARDY-. Both implementations are equivalent since the negated state of this signal will appear the same to the host as the released state.

Since this timing was defined for the sole purpose of requiring DMARDY- to be asserted before IORDY is driven, the minimum value for this timing in all modes is 0 ns.

### E.5.2.27 Setup and hold before DMACK- time ($t_{ACK}$)

The $t_{ACK}$ value is defined for the setup and hold times before assertion and after negation of DMACK-. It is applied to all control signals generated by the host related to an Ultra DMA burst. These signals are STOP, HDMARDY-, HSTROBE, CS1-, CS0-, and DA(2:0). The burst begins with the assertion of DMACK- and ends with the negation of DMACK-. For this burst period, all control signals start, remain, and end in specific states as defined by the protocol. Since there may be some skew between signals from the host to the device due to transmission and component I/O circuitry effects, the host is required to set up all the control signals before asserting DMACK-. This insures that by the time all the signals reach the device, they will all be in the proper state when DMACK- is asserted. Using $t_{ACK}$ as the hold time for the signals after the negation of DMACK- insures that at the termination of the burst, the control signals as seen by the device are in the proper states. This avoids any device state machine confusion. Based on the same analysis used for $t_{MLI}$, the minimum for $t_{ACK}$ is 20 ns.

### E.5.2.28 STROBE to DMARQ/STOP time ($t_{SS}$)

This is the minimum time after a STROBE edge before a device as a sender negates DMARQ or a host as a sender asserts STOP to terminate a transfer. This time is to allow at least one recipient clock cycle between the last STROBE and the termination signal to avoid the possibility of a race condition between the two events and ensure the last word is seen as valid by the recipient. The formula used to determine $t_{SS}$ minimum is:

+ Sender's component I/O to recipient's component I/O actual thresholds max positive skew
+ Max input skew
+ (One recipient clock cycle at the maximum period due to frequency variation) $*$ (clock cycle time)

For modes 0 and 1, a 25 MHz recipient clock is assumed and for all other modes a 30 MHz recipient clock is assumed. While the value specified could have been lower for modes using 30 MHz or higher clock frequencies, $t_{SS}$ is specified to be the same value for all modes for extra margin.

### E.5.2.29 Data setup time at IC component ($t_{DSIC}$) (modes higher than 4 only)

This parameter defines the minimum setup time at the input to the recipient's IC given linear 0.4 V/ns transitions on DD(15:0) and STROBE through 1.5 V. It is included so that IC designers will have an explicit recipient setup time to be simulated during the design phase that will result in a functional system. The formula and details used to determine $t_{DSIC}$ are identical to the ones used to determine $t_{DS}$ above except for the following. First, the IC-to-IC skew is used rather than IC-to-connector skew as in $t_{DS}$. Second, since the IC-to-I/C skew is defined from 1.5 V at the sender to the actual thresholds of the recipient's I/C, the value of the maximum input skew from 1.5 V to actual thresholds with linear 0.4 V/ns input defined in IC input skew above is added to the modified $t_{DS}$ value. The resulting value is the setup time with 0.4 V/ns transitions through 1.5 V that shall be met to be equivalent to the setup time generated in a functioning system.

### E.5.2.30 Data hold time at IC component ($t_{DHIC}$) (modes higher than 4 only)

This parameter defines the minimum hold time at the input to the recipient's IC given linear 0.4 V/ns transitions on DD(15:0) and STROBE through 1.5 V. It is included so that IC designers will have an explicit recipient hold time to be simulated during the design phase that will result in a functional system. The formula and details used to determine $t_{DHIC}$ are identical to the ones used to determine $t_{DH}$ except for the following. First, the IC-to-IC skew is used rather than IC-to-connector skew as in $t_{DS}$. Second, since the IC-to-IC skew is defined from 1.5 V at the sender to the actual thresholds of the recipient's IC, the value of the maximum input skew from 1.5 V to actual thresholds with linear 0.4 V/ns input defined in IC input skew above is added to the modified $t_{DS}$ value. The resulting value is the hold time with 0.4 V/ns transitions through 1.5 V that shall be met to be equivalent to the hold generated in a functioning system.

#### E.5.2.31 Data valid setup time at IC component ($t_{DVSIC}$) (modes higher than 4 only)

This parameter defines the minimum setup time that the sender must generate at the I/O pin into a defined load in order to meet all setup times in a system. It is included so that IC designers will have an explicit sender setup time to be simulated during the design phase that will result in a functional system. The formula and details used to determine $t_{DVSIC}$ are identical to the ones used to determine $t_{DVS}$ except for the following. First, the maximum PCB trace skew, the maximum falling source transition delay, and minimum rising source transition delay are removed from the equation because they account for the skews that occur after the IC. Second, since the maximum output skew includes noise on the signal seen in a functional system that is usually not included in a simulation of an I/O into a lumped load, the noise skew defined above is added to the modified $t_{DVS}$ value. The resulting value is the setup time at the sender IC that shall be met to be equivalent to the setup requirements for a system.

#### E.5.2.32 Data valid hold time at component IC ($t_{DVHIC}$) (modes higher than 4 only)

This parameter defines the minimum hold time that the sender component must generate at the I/O pin into a defined load in order to meet all hold times in a system. It is included so that IC designers will have an explicit sender hold time that could be simulated during the design phase that will result in a functional system. The formula and details used to determine $t_{DVHIC}$ are identical to the ones used to determine $t_{DVH}$ except for the following. First, the maximum PCB trace skew, the maximum falling source transition delay, and the minimum rising source transition delay are removed from the equation because they account for the skews that occur after the component I/O. Second, since the maximum output skew includes noise on the signal seen in a system that would not be part of a simulation of and I/O into a lumped load, the noise skew defined above is added to the modified $t_{DVH}$ value. The resulting value is the hold time at the sender component I/O that shall be met to be equivalent to the hold requirements for a system.

### E.5.3 Ultra DMA Protocol Considerations

#### E.5.3.1 Recipient pauses

The Ultra DMA protocol allows a recipient to pause a burst at any point in the transfer. The clauses below discuss some of the issues and design considerations associated with the Ultra DMA recipient pausing protocol.

##### E.5.3.1.1 DMARDY- minimum negation time

An Ultra DMA recipient pause is initiated through the recipient's negation of DMARDY-. Once DMARDY- is negated, the protocol allows for additional words to be transferred. Pausing is typically done for two reasons. One is that the recipient’s input FIFO or buffer is almost full and would overflow if the burst continued. The second is that the recipient is preparing to terminate the burst. Normally the case of pausing to free space in the FIFO or buffer would result in DMARDY- being negated for at least a few transfer cycles. However, there is no minimum time for the negation of DMARDY-. The recipient does not have to wait for possible additional words or for any minimum time from when the recipient negates DMARDY- until it re-asserts DMARDY-. If, after negating DMARDY-, the recipient becomes ready, it may immediately reassert DMARDY-. Based on the implementation of the sender, a negation and immediate re-assertion of DMARDY- may cause a subsequent STROBE to be delayed. It is recommended that some hysteresis be used in the FIFO trigger points for assertion and negation of DMARDY- to avoid DMARDY- being negated after every word or two.

##### E.5.3.1.2 Number of additional words from sender

An Ultra DMA burst may be paused with zero, one, or two additional data transfers as seen at the recipient connector for modes 0, 1, and 2, and up to three additional transfers for modes 3, 4, 5, and 6. This does not imply that the sender is allowed to send up to two or three more STROBES after it detects the negation of DMARDY-. In most cases it would be a violation of $t_{RFS}$ to do so. Rather than counting words after detecting the negation of DMARDY-, under all conditions the sender stops generating STROBE edges within $t_{RFS}$ of the recipient negating DMARDY-. Even in cases where $t_{RFS}$ is met and less than the maximum number of words are sent, it is still possible for the recipient to see the maximum number of STROBE edges after it negates

DMARDY-. This is due to the delay of the signals through the cable. An example of this is explained below and shown in figure E.31.

In mode 2 when the STROBE time is 60 ns and signal delays add up to 6 ns, both STROBE from sender to recipient and DMARDY- from recipient to sender experience a cable delay of 6 ns. While the recipient negates DMARDY- after the sender toggles STROBE, it does not receive the STROBE transition until after the DMARDY- negation. This would account for the first word received. By the time the sender detects the DMARDY- negation, there are only 49 ns until the next STROBE. This STROBE is within $t_{RFS}$ so the sender may send STROBE without violating the protocol. To the recipient, this would be the second transfer after it has negated DMARDY-, but to the sender it would be the first and only allowable STROBE transition after detecting the DMARDY- negation.

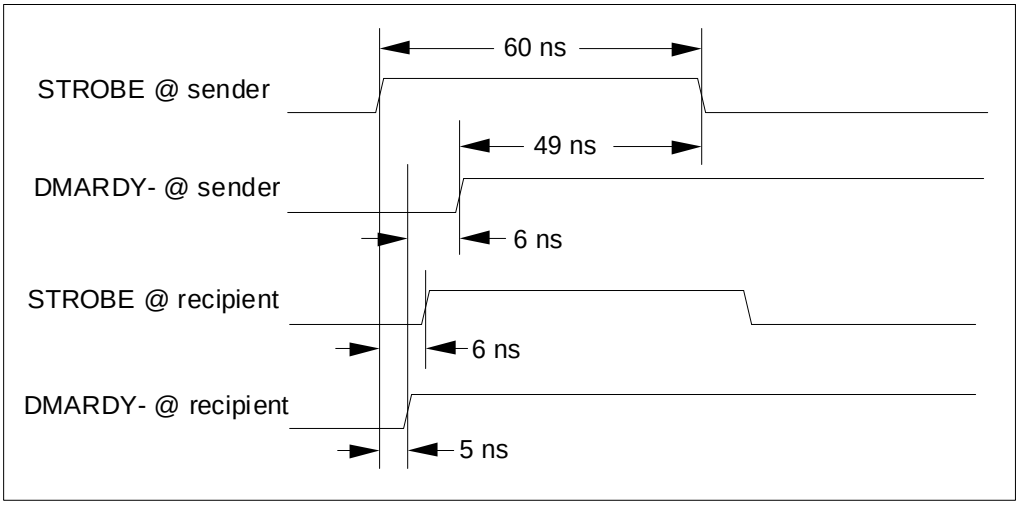



**Figure E.31 - STROBE and DMARDY- at sender and recipient**

#### E.5.3.1.3 Sender output data handling during a pause

In most cases of a recipient pause, a sender stops toggling STROBE in less than one transfer cycle time after DMARDY- negates at its input in order to meet $t_{RFS}$. Since the incoming DMARDY- negation is asynchronous with the sender's internal clock, the incoming DMARDY- signal should be synchronized with the internal clock. In this condition data may be gated or latched to DD(15:0) but never strobed.

If an output register is used when data is transferred from memory for presentation on DD(15:0), no assumptions are made that that data has been or will be transferred. If a pointer in memory is incremented or the data is cleared from memory when it is sent to the output register, data may be lost unless some recovery mechanism is present to decrement the pointer or restore the data if it is never strobed due to a burst termination after a pause. During a pause, other bus activity like a Status register read might occur. A design using an output register would have data in that register overwritten during this other activity. Other designs may involve similar considerations. It is most important to remember that data on DD(15:0) is not sent and is not to be treated as sent until there is a valid STROBE edge.

#### E.5.3.1.4 Additional words at recipient

After DMARDY- is negated, the recipient may receive additional data words. There will be some output delay of DMARDY- from the logic that first generates it inside the IC to the connector, and there will be input delay of STROBE from the connector to inside the IC. In addition to this, data may be pipelined before the FIFO and there may be logic delays between triggering a near full condition in the FIFO and generating the DMARDY- negation. The depth of the recipient's input FIFO where it triggers a condition to negate DMARDY- to avoid an overflow is therefore dependent on the particular design approach. When determining the FIFO trigger point, all FIFO near-full trigger threshold-to-DMARDY- negation delays, the cable delay, $t_{RFS}$ time, input delays, input data pipelining, and the minimum cycle time for the mode supported should be considered.

The recipient may receive STROBE edges until $t_{RP}$ after it negates DMARDY-. The receipt of two or three words by the recipient after a pause has been initiated is not an indication that the sender has paused. The recipient waits until $t_{RP}$ after the pause was initiated before taking any other action (e.g., terminating the burst). Waiting $t_{RP}$ allows for cable delays between the recipient and sender and allows the sender time to complete its process of transitioning to a paused state. The process of switching to a paused state may take additional system clocks after the sender has sent it's last STROBE transition.

Since the recipient's and sender's clocks are asynchronous with respect to each other, there is not a single fixed number of words that the recipient will receive after negating DMARDY-. Every time a recipient begins a pause, a sender may send from zero to the maximum number of words specified for the mode. The Ultra DMA protocol does not give the recipient any means of pausing or stopping on an exact, predetermined boundary.

#### E.5.3.2 CRC calculation and comparison

For each STROBE transition used for data transfer, both the host and device calculate a CRC value. Only words successfully transferred in the transfer phase of the burst are used to calculate this value. This includes words transferred after a pause has been requested. Words put on DD(15:0) but never strobed are not to be used for CRC calculation. In addition, if STROBE is negated at the end of a pause and then the burst is terminated, the protocol requires STROBE to be re-asserted after DMARQ is negated or STOP is asserted. No data is transferred on this STROBE edge and any data on DD(15:0) that was not strobed during the transfer phase of the burst is not used in the CRC calculation on this re-assertion of STROBE.

It is not advisable to use STROBE to clock the CRC generator. Noise on the STROBE signal could cause the recipient's CRC generator to double-clock the generator on a single edge. At the same time, the noise glitch seen by the CRC generator may not affect the data input portion of the logic. This type of implementation has led to CRC errors on systems where data is properly received but the wrong CRC value is calculated. Using different versions of STROBE to clock the CRC generator and to clock data into the FIFO or buffer also leads to a fatal error that has been seen on an implementation of the Ultra DMA protocol. Noise, lack of setup or hold time, race conditions in the logic, or other problems could result in the wrong data being clocked into the FIFO or buffer. At the same time the correct data may be clocked into the CRC generator since it is using a different instance of STROBE. In this case, the resultant CRC value is correct when the data in the recipient is not.

Designs may internally generate a delayed version of STROBE that is synchronous with the recipient clock. This synchronized version of the STROBE is then used to place data into the FIFO or buffer. It is advisable for the recipient to use the same clock that places data into its FIFO or buffer to clock data into its CRC generator. Following this design approach will maximize the probability of clocking the same data into both the CRC generator and FIFO or buffer and clocking both the same number of times.

The standard includes the equations that define the XOR manipulations to make on each bit and the structure required to perform the calculation using a clock generated from STROBE. Through the given equations, the correct CRC is calculated by using a small number of XOR gates, a single 16-bit latch, and a word clock (one clock per STROBE edge). The equations define the value and order of each bit, and the order of each bit is mapped to the same order lines of DD(15:0). The CRC register is pre-set to 4ABAh. This requires pre-setting the latch (CRCOUT) to 4ABAh before the first word clock occurs. After that, CRCIN15 to the latch is tied through to CRCOUT15. When the burst is terminated CRCOUT15 is the final CRC bit 15 that is sent or received on DD15. This direct matching of bit order is true for all CRC bits. The proper use of the data sent on DD(15:0) during the burst transfer is defined in the equations. The value on the DD15 signal line has the same value as bit DD15 in the equations to calculate CRC. This direct mapping is true for all bits strobed on DD(15:0) during a burst.

Once the burst is terminated and the host sends the CRC data to the device (the host always sends the CRC independent of whether the burst was a data-in or data-out transfer), the device compares this to the CRC it has calculated. While other CRC validation implementations may be possible, a CRC input register may be used on the device in combination with a digital comparator to verify that the CRC value in the input register matches the value in its own CRC calculation register.

The longer the Ultra DMA burst size the greater the likelihood of an undetected error. Burst lengths longer than 131,072 bytes will increase the probability of undetected errors above that of the ATA/ATAPI-6 and earlier standards.

#### E.5.3.3 The IDENTIFY DEVICE and IDENTIFY PACKET DEVICE commands

A device communicates its Ultra DMA capabilities and current settings to the host in the data returned by the device as a result of an IDENTIFY DEVICE or IDENTIFY PACKET DEVICE command.

For the PIO and Multiword DMA protocols, only the host generates data STROBES so the minimum cycle times reported for those protocols in the IDENTIFY DEVICE and IDENTIFY PACKET DEVICE data are used by the host for both data-in and data-out transfers to insure that the device's capabilities are not exceeded. For the Ultra DMA protocol, both the host and device strobe data depending on the direction of the transfer. The host determines a mode setting based on both the device's capabilities and its own. The sender may send data (toggle STROBE) at a minimum period of $t_{CYC}$. A recipient receives data at the minimum $t_{CYC}$ for the currently active mode. If the device indicates that it is capable of an Ultra DMA mode, it receives at the minimum time for that mode, no additional cycle time information is required.

#### E.5.3.4 STROBE minimums and maximums

The Ultra DMA protocol does not define a maximum STROBE time. The sender may strobe as slowly as it chooses independent of the mode that has been set, though it has to meet the specified setup and hold times for the mode that has been set by the host. The sender is also not required to maintain a consistent cycle time throughout the burst. It would not be a violation of protocol for the cycle time to change on every cycle so long as all cycles are longer than or equal to the minimum cycle time for the mode that is set. An upper timing bound or PLL is not used by the recipient to qualify the STROBE signal. Regardless of the frequency of the STROBE, the recipient has to meet the setup and hold times of the received signal specified for the mode that has been set. The limit on the maximum STROBE time is determined by the system time-out. This time-out is typically on the order of a few seconds. If a device begins to strobe once every ten seconds during a data-in burst, this would not be in violation of the protocol. However, this could cause a software driver to determine that the device is not responding and perform a recovery mechanism. The recovery will often be a hardware reset to the device.

Unlike a recipient pause where the recipient has to wait $t_{RP}$ after negating DMARDY- before the pause is complete. The sender may consider the burst paused as soon as it meets the data hold time $t_{DVH}$. The implication of this is that data to the recipient may stop on any word. After each word, the recipient waits (with exception of pauses or stops) but never requires an additional word before allowing the burst to be terminated.

#### E.5.3.5 Typical STROBE cycle timing

Neither minimum nor typical cycle times are required to be used by the sender. Other cycle times may be used by systems that do not have internal clocks that provide a frequency to generate signals at those cycle times. The typical mode 1 cycle time of 80 ns will not be met using a common system clock rate of 66.7 MHz. Instead a STROBE cycle time of 90 ns for mode 1 is used and is not a violation of the specification. A typical cycle time of 90 ns reflects 22.2 megabytes per second.

#### E.5.3.6 Holding data to meet setup and hold times

Following are three examples of holding data in an attempt to meet the setup and hold times. The first method is to use the same clock edge to change data and the STROBE, but delay the data through some gates. The second method is to use one edge of the clock to change the STROBE and then use the next opposite edge to change data (half cycle). The third method is to use one active edge of the clock to change STROBE and the next to change data.

Using gate delays to hold data may lead to large variations in hold time over process, temperature, and supply voltage. Meeting Ultra DMA mode 4 or higher timings with gate delays to hold data is not advisable and could lead to timing violations under some conditions. Mode 4 hold time may be met by a single

66.6MHz clock cycle with all timings being met. With a slower 25, 30, or 33 MHz clock, a half cycle rather than full cycle hold would be required in order to still meet the setup time requirements for the higher modes. If the data transitions are not at the middle of a mode 4 or higher cycle, either the setup or hold time margin will be reduced.

#### E.5.3.7 Opportunities for the host to delay the start of a burst

After a device has asserted DMARQ, the host has one opportunity to delay the start of the burst indefinitely for a data-in burst and two opportunities for a data-out burst. For both a data-in and a data-out burst, the first opportunity that the host has to delay the burst is by delaying the assertion of DMACK-. This delay has no specified maximum limit. This is necessary for cases of overlap in system bus access that may cause a delay in the time it takes for the host to become ready to receive data from a device after sending a data-in command. For a data-out burst, the host may delay the first STROBE signal. The difference in overhead between delaying and not delaying may seem small but may still be used to optimize for a faster overall system data transfer rate. The device does not delay its STROBE indefinitely since the device controls the signal that starts the transfer process (DMARQ).

Note that it is a violation of the protocol to terminate the burst unless at least one word has been transferred. After asserting DMACK- the host sends or receives at least one word of data before terminating a burst.

#### E.5.3.8 Maximums on all control signals from the device

The timings for all signals from the device used to perform burst initiation, pause, and burst termination have maximum values. This is to bound the time it takes to perform burst initiation, pause, and termination so the host always knows in advance how long tasks performed by the device may take. Rather than waiting a few seconds for a command or burst to time-out, the host determines that a problem exists if activity is not detected within the specified maximums and sets time-outs for functions performed by the device. For instance, the longest the initiation of a data-in burst may take from the host assertion of DMACK- to the first STROBE is $t_{ENV}$ max plus $t_{RS}$ max. Also, the host may require a burst to terminate in a timely manner in order to service some other device on the bus or the system depending on the chip set design.

## E.6 Cable detection

The ATA interface was originally designed to use a 40-conductor cable assembly. To achieve transfer rates for Ultra DMA modes greater than 2 an 80-conductor cable assembly was developed. This assembly improves signal quality allowing for the faster transfer rates. The improvement in signal quality provided by the 80-conductor cable assembly is the result of placing ground lines between each of the 40 signal lines to decrease crosstalk and reduce settling times. In order to use Ultra DMA modes greater than 2, hosts are required to determine that an 80-conductor cable assembly is installed in the system. The methods for performing this detection are described in the standard. The following clauses provide additional information about determination of cable assembly type.

### E.6.1 80-conductor cable assembly electrical feature

In order to allow detection of the presence of an 80-conductor cable assembly by a host, a unique electrical configuration for the 80-conductor cable assembly was developed. For the 40-conductor cable assembly, PDIAG- is connected to the associated pin on all three connectors. For the 80-conductor cable assembly, PDIAG- was opened between the device connectors and the host connector and grounded in the host connector. Since PDIAG- is pulled up through resistors on devices, this signal will be electrically high at the host connector pin in a system with a 40-conductor cable assembly, and will be at ground at the host connector pin in a system with an 80-conductor cable assembly. Therefore, the host can determine the cable assembly type by sensing the electrical state of this pin. PDIAG- was renamed to be PDIAG-:CBLID- to reflect this additional function. PDIAG- remains connected between the two devices in the 80-conductor cable assembly for proper power-on and hardware reset handshaking.

### E.6.2 Host determination of cable assembly type

In a system with an 80-conductor cable assembly, devices cannot affect the state of CBLID- at the host connector since that signal is open to the host and is grounded inside the host connector. In a system with a 40-conductor cable assembly, the state of PDIAG- at the host will be the same at all three connectors. Since the devices are required to have pull-up resistors on PDIAG-, the state of that signal will be high when all devices have released it. The standard specifies that devices shall release PDIAG- after the first command has been sent to Device 1 after a power-on or hardware reset. If, after that, the host senses the signal as electrically high, a 40-conductor cable assembly is installed in the system. If the host senses the signal as electrically low, then there are two possibilities: either an 80-conductor cable assembly is installed in the system, or a 40-conductor cable assembly is installed in the system and Device 1 is continuing to assert PDIAG- after is should have released that signal.

Any device claiming compliance to ATA-3 or later as indicated in IDENTIFY DEVICE or IDENTIFY PACKET DEVICE data should properly release PDIAG- after a power-on or hardware reset upon receiving the first command or after 31 seconds have elapsed since the reset, whichever comes first. This is why the standard specifies that one of those two commands is to be sent to Device 1 before a host samples the state of CBLID-. From the returned data the host can determine that device 1 indicates compliance with the standard, and, if it does, that the device has released PDIAG-.

An advantage of this detection method is that the cable assembly type may be determined by the host regardless of the devices attached to the cable. Two disadvantages are: this method does require an IC pin on the host for each port to connect to CBLID-, and, as described above, it is possible for a host to make an incorrect determination of cable type if a device is present that does not correctly release PDIAG-.

### E.6.3 Device determination of cable assembly type

A second method for determination of cable assembly type was developed because some hosts capable of operating at Ultra DMA modes greater than 2 were unable to connect to CBLID-, even though this is required by the standard. For this method the host is required to place a capacitor on CBLID- to ground. After receiving an IDENTIFY DEVICE or IDENTIFY PACKET DEVICE command a device negates PDIAG-:CBLID- allowing the capacitor (if present) to discharge, releases PDIAG-:CBLID-, samples PDIAG-:CBLID- within a window where the signal would still be low before the capacitor charges, and then returns data for the command. If the device senses the signal as electrically low, then there are two possibilities: either a 40-conductor cable assembly is installed in the system, or an 80-conductor cable assembly is installed in the system and Device 1 is continuing to assert PDIAG- after is should have released that signal. If the device senses the signal as electrically high, then an 80-conductor cable assembly is installed in the system. The device places the results from its determination of cable assembly type into the data returned for the IDENTIFY DEVICE command.

As with the host determination method, the host uses the data from the IDENTIFY DEVICE or IDENTIFY PACKET DEVICE command from both devices to verify that Device 1 indicates compliance with ATA-3. The host then checks to see what cable assembly type the device indicates having determined. For this algorithm the host issues the IDENTIFY DEVICE or IDENTIFY PACKET DEVICE command to Device 1 first to insure that it has properly released PDIAG-.

Two advantages to device determination of cable assembly type are, first, the host is not required to use an IC pin for PDIAG- on each port that supports Ultra DMA modes greater than 2. The device assisted cable detection method provides the only solution for this configuration. Second, while both the host and device cable assembly type determination methods may result in incorrect determination of cable assembly type when Device 1 does not correctly release PDIAG-, Device 0 will indicate that it has determined that a 40-conductor cable assembly is installed when an 80-conductor cable assembly is installed. Though the host will not set Device 0 to operate at Ultra DMA modes greater than 2, this will not result in the potential for poor system operation that could result if a host set a device to operate at Ultra DMA modes greater than 2 when a 40-conductor cable assembly is installed in a system.

#### E.6.3.1 Capacitor on CBLID-

All devices must be able to assert PDIAG-. This is required for the device to function as Device 1 during execution of the power-on hardware and software reset protocols. The standard recommends an open collector output on that pin. Some designers may chose to use drivers capable of a high-impedance state for this signal but never negate it. The signal is negated to a high state through a required 10 kΩ pull-up resistor on each device. All devices must also be able to detect the state of PDIAG-. This is required for the device to function as Device 0 during execution of the power-on hardware and software reset protocols. The timing for asserting or detecting PDIAG- power-on hardware and software reset protocols is on the order of milliseconds or more so the assertion and detection need not be controlled by hardware only but rather be controlled through firmware or some combination of hardware and firmware. In order for the device detection algorithm to work properly, the device firmware needs to have I/O control of PDIAG-.

##### E.6.3.1.1 Capacitor size determination

The device pull-up on PDIAG- must always be to 5 V rather than 3.3 V as is required for all other signals for Ultra DMA mode 5. On some hosts a 10 kΩ pull-down resistor was placed on PDIAG-:CBLID- so that the signal could be connected to a non-5 V tolerant input. In this case, the pull-down resistor acts in conjunction with the device pull-up(s) to divide the PDIAG-:CBLID- voltage. If a 3.3 V pull-up is used, the value of CBLID- detected by the host input could be only 1.5 V, which is not a valid high level. The determination of capacitor size is based on a pull-up to 5 V.

With the capacitor installed, the power-on and reset handshaking must still function. At the beginning of this handshaking, Device 1 releases PDIAG- within 400 ns after reset, and the signal must be electrically high within 1 ms after reset. While a single Device 1 configuration is defined in the standard, this is an atypical configuration. However, the timing for handshaking during the power-on and reset protocols for all configuration must be taken into account when selecting a capacitor value. The pull-up resistors on the

devices are specified to be 10 kΩ plus or minus 5%. In addition, a device may have up to 20 pF of load on any signal, the host may have up to 25 pF of load on any signal, and the cable may add an additional 40 pF to any signal.

V = threshold voltage
$V_{PU}$ = pull-up voltage
C = Capacitive load on signal
R = pull-up resistor value
t = time to reach threshold

$V = V_{PU} ( 1 - e^{( -t / RC )} )$ assuming that the starting voltage is 0 V
Solving for C: $C = -t / (R ( \ln ( 1 - ( V / V_{PU} ) ) ) )$
Solving for t: $t = ( -R * C ) * ( \ln (1 - ( V / V_{PU} ) ) )$

As described above, there are host systems that place a 10 kΩ pull-down resistor on PDIAG-:CBLID- at the host. The worst-case condition for slow pull-up on PDIAG- would be with a +5% resistor on the device, a –5% resistor at the host, and a –10% supply voltage. The Thevenin equivalent of this voltage divider circuit given these values is the following:

$V_{iH}$ = 2.0 V
$V_{PU}$ = 2.14 V
R = 4990 Ω
t = 1.0 ms

Using the equation for C and the values immediately above, the maximum value that may be used for C is 0.073 µF. With a higher value, the timing for handshaking may no longer be met. A lower value shortens the time to charge the capacitor by the device pull-up resistors making the detection window narrower. A standard value capacitor is 0.047 µF. With a 20% tolerance, the maximum value would be 0.0564 µF, which is within the maximum limit. The additional loading of 20, 25, and 40 pF for the device, host, and cable as mentioned above is insignificant given this order magnitude capacitor.

**E.6.3.1.2 Timing window determination**

Given the selection of a 0.047 µF capacitor, the limits of the window where PDIAG- would be high for an 80-conductor cable and low for a 40-conductor cable may be determined.

The minimum time of the window is determined by the maximum time it will take for PDIAG- to go to an electrically high state with an 80-conductor cable assembly installed in a system. With an 80-conductor cable assembly installed, the capacitor would not be loading the PDIAG- signal because the signal is not connected from the host to the devices. The only load would be that of the device itself (20 pF maximum) and the cable (40 pF maximum). Since the load of the cable is independent of the number of devices attached, the maximum rise time will be for the case with a single device installed on the cable. As described above, the pull-up resistor is connected to 5 V in order to be compatible with some hosts. The pull-up resistor could be 10 kΩ +5%.

R = 10500 ohm
$V_{pu}$ = 4.5 V
$V_{iH}$ = 2.0 V
C = 60 pF

With the above values and the time equation given in E.6.3.1.1, the longest time that PDIAG- may take to reach the electrically high state in a system with an 80-conductor cable assembly installed is 0.3 µs.

The maximum time of the window is determined by the minimum time that it will take for PDIAG- to an electrically high state with a 40-conductor cable assembly installed in a system. With a 40-conductor cable assembly installed, the capacitor at the host is connected to the devices through the cable. Since each device has a pull-up resistor, the shortest time for the signal to go to an electrically high state will occur when

two devices are installed on a cable. Each device will have a pull-up resistor as low as 9.5 kΩ (10 kΩ minus 5%). Additionally, some devices have pull-up current through their IC. The highest additional IC pull-up current may be equivalent to a 27500 Ω resistor. The two external and two internal pull-up resistors in parallel are equivalent to a single 3530 Ω resistor. An ATA output driver must be able to drive $V_{oL}$ of 0.5 V at $I_{oL}$. To reach 0.5 V with an $I_{oL}$ of 4 mA, the driver must have a resistance to ground of less than 125 Ω. At 125 Ω, a driver would pull-down a 3530 Ω load to 188 mV (i.e. 5.5 ∗ 125 ∗ ( 3530 + 125 ) ). Assume though that the voltage is only pulled to 0.3 V before it is released.

R = 3530 Ω
$V_{PU}$ = 5.5 V
C = 0.0376 µF (lowest value for 20% tolerance capacitor)

With V = 0.8 V: t = 20.4 µs
With V = 0.3 V: t = 7.4 µs

With the above values, the shortest time that PDIAG- may take to reach an electrically high state with a 40-conductor cable (from 0.3 to 0.8 V) is 13 µs.

It is also important to know how long it will take to discharge the capacitor when a device asserts PDIAG-. The maximum time to discharge the capacitor (down to 0.3 V) would depend on the maximum resistance to ground. As stated above, the driver itself may have 125 Ω maximum. In combination with the 3530 Ω pull-up, the lowest level that the signal could reach is 188 mV. Assuming this were 200 mV, the exponential curve will be to a minimum of 200 mV. This would be equivalent to a curve from 5.3 to 0 V.

$V = V_{PU} (e^{(-t / RC)})$
Solving for t: $t = ( -R * C ) * ( \ln ( V / V_{PU} ) )$

V = 0.1 V
$V_{PU}$ = 5.3 V
R = 125 Ω
C = 0.0564 µF (maximum for 20% tolerance capacitor)

These numbers result in a maximum 28 µs to discharge the capacitor.

# Annex F
(informative)
## Register selection address summary

Table F.1 summarizes the selection addresses for registers for all except the PACKET and SERVICE commands. Table F.2 summarizes the selection addresses for registers for the PACKET and SERVICE commands.

**Table F.1 – Register functions and selection addresses except PACKET and SERVICE commands**

| Addresses | | | | | Functions | |
|---|---|---|---|---|---|---|
| **CS0-** | **CS1-** | **DA2** | **DA1** | **DA0** | **Read (DIOR-)** | **Write (DIOW-)** |
| N | N | x | x | x | Released | Not used |
| | | | | | **Control block registers** | |
| N | A | N | x | x | Released | Not used |
| N | A | A | N | x | Released | Not used |
| N | A | A | A | N | Alternate Status | Device Control |
| N | A | A | A | A | Obsolete(see note) | Not used |
| | | | | | **Command block registers** | |
| A | N | N | N | N | Data | Data |
| A | N | N | N | A | Error | Features |
| A | N | N | A | N | Sector Count | Sector Count |
| A | N | N | A | A | LBA Low | LBA Low |
| A | N | A | N | N | LBA Mid | LBA Mid |
| A | N | A | N | A | LBA High | LBA High |
| A | N | A | A | N | Device | Device |
| A | N | A | A | A | Status | Command |
| A | A | x | x | x | Released | Not used |

Key:
A = signal asserted N = signal negated x = don't care
NOTE – This register is obsolete. It is recommended that a device not respond to a read of this address.

**Table F.2 – Register functions and selection addresses for PACKET and SERVICE commands**

| Addresses | | | | | Functions | |
|---|---|---|---|---|---|---|
| **CS0-** | **CS1-** | **DA2** | **DA1** | **DA0** | **Read (DIOR-)** | **Write (DIOW-)** |
| N | N | x | x | x | Released | Not used |
| | | | | | **Control block registers** | |
| N | A | N | x | x | Released | Not used |
| N | A | A | N | x | Released | Not used |
| N | A | A | A | N | Alternate Status | Device Control |
| N | A | A | A | A | Obsolete(see note) | Not used |
| | | | | | **Command block registers** | |
| A | N | N | N | N | Data | Data |
| A | N | N | N | A | Error | Features |
| A | N | N | A | N | Interrupt reason | |
| A | N | N | A | A | | |
| A | N | A | N | N | Byte Count low | Byte Count low |
| A | N | A | N | A | Byte Count high | Byte Count high |
| A | N | A | A | N | Device select | Device select |
| A | N | A | A | A | Status | Command |
| A | A | x | x | x | Released | Not used |

Key:
A = signal asserted N = signal negated x = don't care
NOTE – This register is obsolete. A device should not respond to a read of this address.

## Annex G
(informative)
## Sample code for CRC and scrambling

**(See Volume 3)**

## Annex H
(informative)
## FIS type field value selection

**(See Volume 3)**

## Annex I
(informative)
## Physical layer implementation examples

**(See Volume 3)**

# Annex J
(informative)
# Command processing overview

**(See Volume 3)**

# Annex K
(informative)
# Physical layer test considerations

**(See Volume 3)**