

DECLARATION OF NATHANIEL E FRANK-WHITE

1. I am a Records Request Processor at the Internet Archive. I make this declaration of my own personal knowledge.

2. The Internet Archive is a website that provides access to a digital library of Internet sites and other cultural artifacts in digital form. Like a paper library, we provide free access to researchers, historians, scholars, and the general public. The Internet Archive has partnered with and receives support from various institutions, including the Library of Congress.

3. The Internet Archive has created a service known as the Wayback Machine. The Wayback Machine makes it possible to browse more than 450 billion pages stored in the Internet Archive's web archive. Visitors to the Wayback Machine can search archives by URL (i.e., a website address). If archived records for a URL are available, the visitor will be presented with a display of available dates. The visitor may select one of those dates, and begin browsing an archived version of the Web. Links on archived files in the Wayback Machine point to other archived files (whether HTML pages or other file types), if any are found for the URL indicated by a given link. For instance, the Wayback Machine is designed such that when a visitor clicks on a hyperlink on an archived page that points to another URL, the visitor will be served the archived file found for the hyperlink's URL with the closest available date to the initial file containing the hyperlink.

4. The archived data made viewable and browseable by the Wayback Machine is obtained by use of web archiving software that automatically stores copies of files available via the Internet, each file preserved as it existed at a particular point in time.

5. The Internet Archive assigns a URL on its site to the archived files in the format http://web.archive.org/web/[Year in yyyy][Month in mm][Day in dd][Time code in hh:mm:ss]/[Archived URL] aka an "extended URL". Thus, the extended URL http://web.archive.org/web/19970126045828/http://www.archive.org/ would be the URL for the record of the Internet Archive home page HTML file (http://www.archive.org/) archived on January 26, 1997 at 4:58 a.m. and 28 seconds (1997/01/26 at 04:58:28). The date indicated by an extended URL applies to a preserved instance of a file for a given URL, but not necessarily to any other files linked therein. Thus, in the case of a page constituted by a primary HTML file and other separate files (e.g., files with images, audio, multimedia, design elements, or other embedded content) linked within that primary HTML file, the primary HTML file and the other files will each have their own respective extended URLs and may not have been archived on the same dates.

6. Attached hereto as Exhibit A are true and accurate copies of browser screenshots of the Internet Archive's records of the archived files for the URLs and the dates specified in the attached coversheet of each printout.

INTERNET ARCHIVE
archive.org

7. Attached hereto as Exhibit B are true and accurate copies of the Internet Archive's records of the archived files for the URLs and the dates specified in the following table.

Folder	URL
001	https://web.archive.org/web/20030411193107/http://www.serialata.org/collateral/zipdownloads/serialata10a.ZIP
002	https://web.archive.org/web/20061025163622/http:/www.eetchina.com/ARTICLES/2006AUG/PDF/HDMI_Spec_1.3_GM1.pdf?SOURCES=DOWNLOAD

8. I, Nathaniel E Frank-White, declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, under Section 1001 of Title 18 of the United States Code.

DATE: September 10, 2024

Nathaniel E Frank-White

Nathaniel E Frank-White

EXHIBIT A

https://web.archive.org/web/20030411130221/http:/www.serialata.org/collateral/index.shtml

INTERNET ARCHIVE
WayBackMachine
http://www.serialata.org/collateral/index.shtml
Go
DEC APR JUN
53 captures
4 Oct 2002 - 6 Jan 20
2002 11 2003 2004
About this capture

Specs and Design Guides

Specs and Design Guides

Serial ATA II: Extensions to Serial ATA 1.0 Specification
The Serial ATA II specification was released on October 18, 2002. Download spec.

SATA 1.0a Specification
The Serial ATA final 1.0a specification was released on February 4, 2003 and incorporates previous version(s) errata. This is the latest approved specification. Download spec.

SATA 1.0 Design Guides
Design guides are supplemental recommendations that may assist in the design and deployment of Serial ATA technology. They should not be considered specification requirements or design requirements. Download design guides. (April 5, 2002)

Serial ATA 1.0 Adopters Agreement/License. Click here for details.

Come see us at the WinHEC 2003 conference. Click here for details.

https://web.archive.org/web/20030604160118/http://standards.ieee.org/sa-mem/corpmemlist.html

INTERNET ARCHIVE WayBackMachine
http://standards.ieee.org/sa-mem/corpmemlist.html
Go
APR JUN AUG
51 captures
5 Jun 2000 - 5 May 20
2002 04 2003 2004
About this capture

Products/Services
Catalog
Project Status Search
Standards Development Solutions
More...

IEEE Standards On-Line

Membership
Join IEEE-SA
Log In to Member Central

Registration/Certification
OUI
Ethertype
POSIX
ITS Data Registry

Standards Development
Working Groups and Committees
Patents Database
Balloting
Policies and Procedures
Process-at-a-Glance

About IEEE-SA
Board of Governors
Standards Board

News Room
What's New@IEEE

FAQs
Membership
Standards Proccess

IEEE Standards Association Corporate Membership

Join Our Team

Member companies represent a broad spectrum of industries, ranging from power engineering to information technology and telecommunications. Through their collective expertise and wisdom, they contribute to the integrity and value that IEEE standards hold today.

AirFlow Networks, Inc.

Allied Telesis K K
Tokyo, Japan

Battery-Biz, Inc.

Bridgewater Systems Corporation
Ottawa, Canada

Broadcom Corporation
Irvine, CA, US

Canadian Electricity Association
Montreal, Canada

COMPAL Electronics, Inc.
Taipei, Taiwan

Dell Computer Corporation
Round Rock, TX, US

Deutsche Telekom AG
Darmstadt, Germany

Dynapack International Technology Corporation
Taoyuan, Taiwan

Fedco Electronics, Inc.
Fond du Lac, WI, US

Force Computers
Westboro, MA , US

Goepel Electronics LLC
Jena, Germany

Hart InterCivic
Lafayette, CO, US

Hewlett-Packard
Cupertino, CA, US

IBM Corporation
Armonk, NY, US

IEEE-Industry Standards & Technology Org
Piscataway, NJ , US

Inventec Corporation
Santa Clara, CA, US

Lucent Technologies
Holmdel, NJ, US

Metalink Broadband Access
Yakum, Israel

Mitra Imaging Inc.
Waterloo, Ontario, Canada

Motorola Incorporated
Lawrenceville, GA, US

National Semiconductor
Santa Clara, CA, US

National Taiwan University - MediaTek Wireless Research Lab
Taipei, Taiwan

Pacific Broadband Networks
Scoresby, Australia

Matsushita Battery Industrial Co/Panasonic
Osaka, Japan

Quanta Computer Inc.
Tao Yuan Shien, Taiwan

Ramtron International Corp.
Colorado Springs, CO, US

Raza Foundries
San Jose, CA, US

Realtek Semiconductor Corp.
Hsinchu, Taiwan

S2IO Technologies Corporation
Ontario, Canada

Samsung Electronics Company
Austin, TX, US

Sanyo Energy Corporation
San Diego, CA, US

SBC Communications
Austin, TX, US

Siemens AG
Munich, Germany

Simplo Technology Company Ltd.
Hsin Chu Hsien, Taiwan

Solectron Corporation
Singapore

Sony Corporation Core Tech & Network Co
Tokyo, Japan

Tata Consultancy Services
Bombay India

Texas Instruments Incorporated
Dallas, TX, US

Thales Communications
Colombes France

Thermon Manufacturing
San Marcos, TX, US

WildPackets Inc.
Walnut Creek, CA US



Wistron Corporation
Taipei, Taiwan

Youtility
Hudson, NH, US

More information on IEEE-SA corporate membership can be found here or by emailing IEEE-SA.

IEEE Standards Association Corporate Membership
PO Box 1331
445 Hoes Lane
Piscataway NJ 08855-1331
USA

(back to top)

Copyright © 2003 IEEE

(r.rubenstein@ieee.org)
URL: http://standards.ieee.org/sa-mem/corpmemlist.html
(Modified:2-Jun-2003)

https://web.archive.org/web/20040214174215/http://standards.ieee.org/getieee802/802.3.html

INTERNET ARCHIVE
WayBackMachine
http://standards.ieee.org/getieee802/802.3.html
Go
DEC FEB APR
14
2002 2004 2005
365 captures
26 May 2001 - 1 Sep
About this capture

IEEE STANDARDS ASSOCIATION

Get IEEE 802™ Program

About Get IEEE 802™

How to support Get IEEE 802™

Contact for Get IEEE 802™

Terms and Conditions

IEEE Standards Association

IEEE-SA Home

IEEE Standards Online

IEEE Registration Authority

IEEE Standards Development Solutions

iEEE News & Announcements

IEEE 802® Working Groups & Executive Committee Study Group

Home Pages

Register for FREE Newsletters





HugeDomains.com
Shop for Over 350,000 Premium Domains

...Partnering with industry to close the digital divide

IEEE 802.3
CSMA/CD Access Method

Trouble downloading standards? Click here for instructions

IEEE 802.3-2002 IEEE Standard for Information technology--Telecommunications and information exchange between systems--Local and metropolitan area networks--Specific requirements--Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications

For faster download
IEEE 802.3-2002 -- Section One
IEEE 802.3-2002 -- Section Two
IEEE 802.3-2002 -- Section Three

IEEE 802.3ae-2002 IEEE Standard for Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications-Media Access Control (MAC) Parameters, Physical Layer and Management Parameters for 10 Gb/s Operation

IEEE 802.3af-2003 *(Includes portions of IEEE 802.3ae)* IEEE Standard for Information technology--Telecommunications and information exchange between systems--Local and metropolitan area networks--Specific requirements--Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications--Amendment Data Terminal Equipment (DTE) Power via Media Dependent Interface (MDI)

1802.3-2001 IEEE Conformance Test Methodology for IEEE Standards for Local and Metropolitan Area Networks-Specific Requirements-Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications *(Incorporates IEEE Std 1802.3d)*



Click here for instructions.

Back to main page



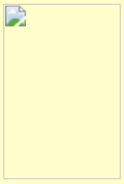
IEEE Standards Zones
Select zone

Related Products

NEW IEEE 802.15.3 Handbook



NEW IEEE 802.15.4 Handbook







Copyright © 2001-2004 IEEE

(e.yearwood@ieee.org)
URL: http://standards.ieee.org/reading/getieee802/802.3.html
(Modified: 5 January 04)

https://web.archive.org/web/20040212061658/http://standards.ieee.org/getieee802/portfolio.html

INTERNET ARCHIVE WayBackMachine
http://standards.ieee.org/getieee802/portfolio.html
Go
DEC FEB APR
12
2003 2004 2005
344 captures
3 Jul 2001 - 3 Sep 20
About this capture

IEEE STANDARDS ASSOCIATION

Get IEEE 802™ Program

About Get IEEE 802™

How to support Get IEEE 802™

Contact for Get IEEE 802™

Terms and Conditions

IEEE Standards Association

IEEE-SA Home

IEEE Standards Online

IEEE Registration Authority

IEEE Standards Development Solutions

News & Announcements

IEEE 802® Working Groups

Home Pages

Register for FREE Newsletters

standards|asia™ Today! Sign up now - it's fast and free!

standards|europe™ Today! Sign up now - it's fast and free!

HugeDomains.com Shop for Over 350,000 Premium Domains

Portfolio to Get IEEE 802™

...Partnering with industry to close the digital divide

Select from our menu to begin downloading...

IEEE 802®: Overview & Architecture

IEEE 802.1™ Bridging & Management

IEEE 802.2™: Logical Link Control

IEEE 802.3™: CSMA/CD Access Method

IEEE 802.4™: Token-Passing Bus Access Method

IEEE 802.5™: Token Ring Access Method

IEEE 802.6™: DQDB Access Method

IEEE 802.7™: Broadband LAN

IEEE 802.10™: Security

IEEE 802.11™: Wireless

IEEE 802.12™: Demand Priority Access

IEEE 802.15™: Wireless Personal Area Networks

IEEE 802.16™: Broadband Wireless Metropolitan Area Networks

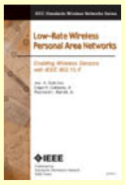
Back to main page



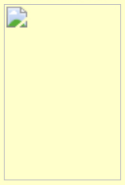
IEEE Standards Zones
Select zone

Related Products

NEW IEEE 802.15.3 Handbook



NEW IEEE 802.15.4 Handbook







Copyright © 2001-2003 IEEE

(e.yearwood@ieee.org)

URL: http://standards.ieee.org/getieee802/portfolio.html

(Modified: 10 November 2003)

https://web.archive.org/web/20030624044640/http://www.standards.ieee.org/getieee802/

INTERNET ARCHIVE WayBackMachine
http://www.standards.ieee.org/getieee802/ Go
665 captures
20 May 2001 - 6 Aug 2
APR 2002 | JUN 24 2003 | AUG 2004
About this capture

IEEE STANDARDS ASSOCIATION

Get IEEE 802™ Program

About Get IEEE 802™

How to support Get IEEE 802™

Contact for Get IEEE 802™

Terms and Conditions

IEEE Standards Association

IEEE-SA Home

IEEE Standards Online

IEEE Registration Authority

IEEE Standards Development Solutions

News & Announcements

IEEE 802® Working Groups & Executive Committee Study Group

Home Pages



Welcome to Get IEEE 802™

...Partnering with industry to close the digital divide

Main page

The *Get IEEE 802*™ program makes IEEE 802® standards available at no charge in PDF format thanks to the sponsors listed to the right. This **pilot program** grants public access to view and download **current** individual IEEE Local and Metropolitan Area Network standards at no charge.

- New IEEE 802® standards are **included in the program after they have been published in PDF for six months.**
- All documents available in the *Get IEEE 802*™ program will remain in the program until they are replaced in the program by a superseding document or are withdrawn.
- Superseded, Drafts, and Withdrawn documents can be purchased by visiting the IEEE Online Catalog & Store, or by subscribing to IEEE LAN/MAN plus Drafts Standards Online Subscription.

This pilot program will be reviewed periodically for continuance. Click here for more information about how you and/or your organization can help keep this pilot program alive.

To select any of the available standards, click here for the index to the standards. You must accept the Terms and Conditions. Upon accepting, you will be able to download the standard(s). If you encounter a problem downloading a standard, try configuring Adobe Acrobat as a helper not a plug-in (see instructions below), or upgrade to Adobe Acrobat 5.0 or higher.

This program may be changed without notice.

Trouble Downloading Standards? Try configuring Adobe Acrobat as a helper not a plug-in. Click here for instructions.



Platinum Level

IEEE 802

Silver Level

Wi-Fi Alliance

*

CISCO Systems

Bronze Level

Applied Micro Circuits Corporation

*

Broadcom Corporation

Contributor

Paul Nikolich



Related Products





IEEE Standards Zones

Select zone

Copyright © 2001-2003 IEEE

(e.yearwood@ieee.org)
URL: http://standards.ieee.org/getieee802/index.html
(Modified: 2003 June 13)

Download the free Shockwave Flash Player.

GET SHOCKWAVE

https://web.archive.org/web/20030804092758/http://www.ieee802.org/3/purchase/index.html

INTERNET ARCHIVE WayBackMachine
http://www.ieee802.org/3/purchase/index.html
Go
JUN AUG FEB
129 captures
27 Dec 2001 - 1 Jul 2(
2002 04 2003 2004
About this capture

Purchasing IEEE 802.3 Standards and Draft Standards

- IEEE 802.3 standards are available, in PDF format, at no charge, six months after their publication:
  - IEEE 802.3 Standards Available for Free Download
- IEEE 802.3 Standards and Draft Standards are available for purchase in both PDF and hardcopy format:
  - IEEE 802.3 Standards available for purchase
  - IEEE 802.3 Drafts available for purchase
- Access to all IEEE 802.3 Standards and Draft Standards is also available by purchasing a Local and Metropolitan Area Networks plus Drafts (LAN/MAN 802s) Standards Subscription:
  - Local and Metropolitan Area Networks Standards Subscription
- Certain portions of the IEEE 802.3 Standard are available in machine readable form (ASCII) for the convenience of purchasers of the standard:
  - IEEE Std 802.3 Machine readable extracts

Return to IEEE 802.3 Home Page
Last Update: 29 Jul 02

search email home ieee corporate

https://web.archive.org/web/20030804072854/http://www.ieee802.org:80/3/af/index.html

INTERNET ARCHIVE WayBackMachine
http://www.ieee802.org/3/af/index.html
Go
106 captures
11 Dec 2000 - 06 May
JUN AUG OCT
2002 04 2003 2004
About this capture

IEEE P802.3af
DTE Power via MDI Task Force

The work of the IEEE P802.3af DTE Power via MDI Task Force is now complete with the approval of IEEE Std 802.3af-2003 at the June 2003 IEEE Standards Board meeting.

IEEE Std. 802.3af-2003 is available from the IEEE..

- IEEE P802.3af Approved Project Authorization Request.
- IEEE DTE Power via MDI Objectives.
- IEEE P802.3af 5 Criteria.

- IEEE P802.3af Requirements document.

- IEEE P802.3af search.
- IEEE P802.3af public area.
- IEEE P802.3af e-mail reflector archive.
- IEEE P802.3af comments received during balloting.

- DTE Power via MDI Study Group public archive area.

Return to IEEE 802.3 Home Page
Last Update: 1 Jul 03

---

search email home ieee corporate

https://web.archive.org/web/20070630082031/http://www.hdmi.org/about/adopters_founders.aspx

INTERNET ARCHIVE WayBackMachine
http://www.hdmi.org/about/adopters_founders.aspx
Go
26 captures
11 Jun 2007 - 1 Sep 2
MAY JUN AUG
2006 30 2007 2008
About this capture

HDMI
HIGH-DEFINITION MULTIMEDIA INTERFACE

Stay Connected
Sign up for updates from HDMI



Adopters & Founders

HDMI Founders

Hitachi, Ltd.
Matsushita Electric Industrial Co., Ltd. (Panasonic)
Philips Consumer Electronics International B.V.
Silicon Image, Inc.
Sony Corporation
Thomson, Inc.
Toshiba Corporation

HDMI Adopters

3 | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z

Last Updated on 6/29/2007

3S Digital Co Ltd
A&R; Cambridge Ltd (Arcam)
Across Techno Inc
Action Star Enterprise Co., Ltd.
Advance & Beyond-Tech Co Ltd
Advanced Digital Broadcast Polska Sp zoo
Advanced Display Lab Inc.
Advanced Knowledge Associates
Advanced Micro Devices
Advanced-Connectek Inc.
Agilent Technologies Inc.
Aiter Cable (Qinfen Aoya)
AJA Video Systems, Inc.
Akai Electric Co Ltd
Al Tech Inc.
Alco Digital Devices Limited
Algolith Inc
ALI Corporation
Allion Computer Inc
Allion Test Labs, Inc.
Alltop Technology Co Ltd
Amco Tec International Inc.
Amino Communications Ltd
Amoi Electronics Co Ltd
Amphenol Corporation
Amphenol East Asia Ltd.
Amstrad PLC
AmTran Technology Co Ltd
AMX
Analog Devices, Inc.
Analogix Semiconductor Inc
Anam Electronics Co Ltd
Anchor Bay Technologies
Antler Computer BV
Apple Computer
ARC Electro Industry Corp.
Archos SA
Arion Technology Inc
Asahi Kasei Microsystems Co Ltd
AstroDesign Inc
Asustek Computer Inc
Atec Co Ltd
Aten International Co Ltd
Atlona
Audio International
Audio Partnership PLC
Audio Technica Corporation
Audiovox Electronics Corporation
Aurora Multimedia Corp
Avid Technology Inc.
Avision Technology (Changzhou) Co., Ltd
Avocent Corporation
BAFO Technologies/TWN 1st Line Elect Cor
Bang & Olufsen
BBK AV Electronics Corp Ltd.
Beko Elektronik A.S.
BenQ Corporation
Better Holdings (HK) Limited
Beyondwiz Co Ltd
Biostar Microtech Int`l Corp.
BizLink Technology
Blackmagic Design
Blue Jeans Cable
Bluevision HK Ltd
Bose Corporation
Broadcom Corporation
Buffalo Inc.

C&C; Technic Taiwan Co Ltd
C.S.L. Wire & Plug Company
CABLE MAX ELECTRONICS CO., LTD
Cable Vision Electronics Co Ltd
Calibre UK LTD
California Micro Devices
Cameo Communications Inc
Canon Inc
Carol Wiring Harness Co Ltd
Cary Audio Design Ltd
Celrun Co Ltd
Chang Yang Electronics Company Ltd
Changzhou Jinyin Futao Photoelectronic Devices Co
Changzhou Wujin Best Electronic Cables Co
Changzhou Wujin Yumin Electronic Co., Ltd
Changzhou Yuyue Electronics Co., Ltd.
Chaoyue Jiangsu Digital Co Ltd
Chau`s Electrical Co. Ltd
Cheertek Inc
Cheng Uei Precision Industry Co Ltd
Chi Lin Technology Co.,LTD.
Chi Mei Optoelectronics Corp
Chi Shing Computer Accessories
China Electronics Technology Limited
Ching Tai Electric Wire & Cable Co Ltd
Chinshong Enterprise Co., Ltd.
Chip Advanced Technology
Chroma ATE, Inc.
Chrontel Inc
Cinemateq GmbH & Co KG
Cinetron Technology Inc.
Circuit Assembly Corporation
Cisco Systems Inc-Linksys
Clipsal Manufacturing Ltd
Coby Electronics Co., Ltd.
Comax Technology Inc.
Comda Advanced Technology Corp
Comoss Electronic Co., Ltd.
Compal Electronics Inc
Conexant Systems, Inc.
Connexions Technology Corp
Conntech Electronic (Suzhou) Corporation
Conrac GmbH
Contac Cable Industrial Ltd
Contek Electronics Co Ltd
Copartner Technology Co Ltd
Coretronic Corporation
CORPO-RIGHT ENTERPRISE CO., LTD
Cosmic Digital Technology Co Ltd
Coulomb Electronics Ltd
Crestron Electronics
CVT International Ltd
Cyberland Co. Ltd
Cypress Semiconductor
Cypress Technology Co Ltd
D&M; Holdings, Inc.
DA2 Technologies Corporation
Daewoo Electronics Corporation
Dalian Golden Hualu Digital Technology Co Ltd
DASAN TPS, Inc.
David Electronics Co Ltd
Delta Electronics Inc.
Der An Electric Wire & Cable Co Ltd
Desay A&V; Science And Technology Co., Ltd
DGSTATION CO., LTD.
DiBoss Co Ltd
Dicon Co. Ltd.
Digeo Interactive LLC
Digi Raum Electronics Limited
Digimedia Technology Co Ltd
Digital Device Inc
Digital Stream Technology Inc.
Digitall World Co Ltd
DigitalZone Co Ltd
Dintek Electronics Ltd.
D-Link Corporation
DM Technology Co Ltd
Dong Guan Asia Tai Electrical Technology
Dong Guan Nissei Electric Co., Ltd.
Dongfangchenming Electronic Co; Ltd.
Dongguan Alllike Electronics Co Ltd
Dongguan City QiRui (KSI-TWI) Electronics Co., LTD
Dongguan Haurtian Wire & Cable Co Ltd
Dongguan Qisheng Electronic Ind
Dongguan Shinko Wire Co Ltd

Dongguan Teconn Connector Co., LTD. (d/b/a LONGKE Electronics)
Dongguan Xinya Electronic Technology Co Ltd
Dtrovision LLC
DVICO Inc.
Eastern Asia Technology Ltd.
Eastern Electronics Co., Ltd
Echostar Technologies Corp
Efficere Technologies
Eizo Nanao Corporation
EliteGroup Computer Systems Co. Ltd.
Elka International Ltd
Ellion Digital Inc
ELSAG S.P.A.
Emine Technology Co Ltd
Energy Master Co Ltd
Enseo Inc
Entery Industrial Co. Ltd.
Entone Technologies (HK) Limited
Enuclia Semiconductor
Erae Electronics
ESS Technology, Inc.
ESSE-GI SRL
Esto Inc
Eternal Fine Technology Development Ltd
Ever Connect Technology Co Ltd
Ever Great Electric Wire and Cable Co Ltd
Ever Much Company Ltd
Evernew Wire & Cable Co Ltd
EVGA Corporation
Explore Microelectronics, Inc.
Extraordinary Technologies Ptd Ltd (Halcro)
Fairchild Semiconductor
Falcon Digital Co., Limited
FCI USA Inc.
First International Computer Inc.
Foosung Tech Corp
Formosa Electric Wire & Cable
Foxconn Electronics d/b/a/ Hon Hai Precision Industry Co., Ltd.
Foxconn Electronics Inc.
Freebox
Freeport Resources Enterprises Corp
FTD Solutions Pte., Ltd.
Fuji Parts Co Ltd
Fujitsu General Limited
Fujitsu Limited
Fujitsu Siemens Computers GmbH
Funai Electric Co Ltd
Galaxy Technology (HK) Ltd.
Gateway Companies Inc
Gearway Electronics Dong Guan Co Ltd
Gefen, Inc.
Genesis Components Inc
Genesis Microchip Inc
Genix InfoComm Co Ltd
Gennum Corporation
Gericom AG
Gigabyte Technology Co Ltd
Global Technologies Inc
Glory Mark Electronic Ltd.
Golana Technology Corporation
Gold Rain Enterprises Corp
Golden Bridge Electech Inc.
Good Mind Industries
Goodwang International Company
Gowell Electronic Ltd
GP Electronics SZ Limited
GPNC Co Ltd
Grandbeing Technology Co Ltd
Greatland Electronics Taiwan Ltd
Greatlink Electronics Taiwan LTD
GREEN-BOX TECHNOLOGY CO. LTD.
Guangzhou Changjia Electronic Co Ltd
H.P.R. Electronics GmbH
Handan BroadInfoCom Co Ltd
Hang Lung Electronics (Hong Kong) Ltd.
Hannspree, Inc.
HannStar Display Corporation
Harman International Industries Inc.
Harsper Co Ltd
High Top Precision Electronic Co. Ltd
Hightech Information System Ltd
Himax Technologies Inc
Hip Hing Cable & Plug Mfy Ltd
Hirakawa Hewtech
Hirose Electric Co Ltd.
Hisense Electric Corp Ltd

Homecast Co Ltd
HongLin Electronic Co Ltd
Horng Tong Enterprise Co Ltd
Hosiden Corporation
Hotron Precision Electronic Ind Co Ltd
HPC Technology Inc
HUB-TECH
Huipu Electronic Shenzhen Co Ltd
Huiyang D&S; Cable Co, Ltd
Huizhou Shenghua Industry Co. Ltd
Humax Co Ltd
Hyosung Electronics Co Ltd
Hyundai Digital Technology Co Ltd
Hyundai ImageQuest Co Ltd
I Plus Technologies Co., Ltd.
i3 Micro Technology ab
IAE Limited
IBEX Technology Co., Ltd.
Idar Electronics Co Ltd
IDK Corporation
Imagenics Co Ltd
Indtek Co Ltd
InFocus Corporation
Info-Tek Corporation
Inkel Corporation
Innovision Multimedia Ltd
Intec, Inc
Integrated Device Technology Inc
Intel Corporation
Intersil Corporation
Inventec Corporation
I-O Data Device Inc
IRI Ubiteq, Inc.
Iriso Electronics Co Ltd
Jabil Circuit Inc
Jalco Co., Ltd.
Jalink Technology Inc.
Japan Aviation Electronics (JAE)
Jean Co Ltd
Jebsee Electronics Co Ltd
Jenving Technology AB / Supra Cables
Jesmay Electronics Co Ltd
Jet Way Information Company Limited
Jhen Vei Electronic Co Ltd
Jiangsu Shinco Electronic Group Co Ltd
JIB Enterprise Group Co Ltd
Ji-Haw Industrial Co Ltd
Jiuh Shiuh(Yun Wah Electron Factory)
Johnson Components & Equipments Co., Ltd.
Joinsoon Electronics MFG, Co LTD
Jye Tai Precision Industrial Co Ltd.
K.S. Terminals Inc.
Kaleidescape, Inc.
Kaonmedia Co Ltd.
Kawasaki Microelectronics Inc
KCS Co Ltd
Kechenda Plastic & Electronic Factory
Keentop International Enterprise Co., Ltd.
KEL Corporation
Kenmark Industrial Co Ltd
Key Digital Systems
Konka Group Co Ltd
Kordz Pty Ltd
Korea Data Systems Co. Ltd
Kramer Electronics Ltd
Krell Industries
KRUEGER & GOTHE GMBH
KTV Global Corp
Kunming Electronics Co., Ltd
KUNSHAN GUOJI Electronics Co; Ltd.
KunZhong Electronic Co Ltd
Kuon Yi Industrial Corporation
Kycon, Inc
Le Hong PO Company Ltd
Leader Electronics Corp.
Leader Wise Technology HK Ltd
Leadtek Research Inc
LeCroy Corporation
Leoco Corporation
LeQing City QiangSheng(EAST) Electronics Co., Ltd.
LG Electronics
LG-Nortel Co. Ltd.
Lin Shiung Enterprise Co Ltd
Link Concept Technology Ltd
Linkiss Co Ltd

Linktec Technologies Co Ltd
Linktech Electronic Products Ltd.
Linn Products Ltd
Lite-On Corporation
Loewe Opta GmbH
Logitech Inc
Longwell Company
Lorom Industrial Co Ltd
Lotes Co Ltd
LSI Logic Corporation
Lumagen, Inc
Lung Hwa Electronics, Co
Luxshare Electronical Ltd
LYNGBOX MEDIA AB
Macro Image Technology Inc
Magnum Semiconductor
Main Super Enterprises Co.
Marketech International Corp.
Marshall Electronics
Marusys Co Ltd
Marvell International Ltd
Maspro Denkoh Corporation
Master Co. LTD-Korea
Master Hill Electric Wire & Cable
Maxim Integrated Products
Media Global Links Co. Ltd
MediaTek Corp.
Meiloon Industrial Co., Ltd.
Meridian Audio Limited
Methode Electronics Inc
Metz-Werke GmbH & Co KG
MG Digital, LLC
Micronas GmbH
Microsoft Corporation
Micro-Star International Co Ltd
Microtek International Inc
MIK21 Co., Ltd
Minwa Electronics Co., Ltd.
Mitsubishi Electric Corporation
Mitsumaru East Kit (Holdings) Ltd.
Mitsumi Electric Co., Ltd.
Mivar Di Carlo Vichi Sapa
Molex Incorporated
Morning Star Industiral Co Ltd
Motorola, Inc.
MSL Enterprises Corp
Mstar Semiconductor, Inc.
Murata Manufacturing Co., Ltd.
Musco Sports Lighting LLC
Mustek International Inc.
NAD Electronics International
Nanjing Wanlida Technology Co Ltd
NEC Display Solutions Inc
NEC Electronics Corporation
NEC Personal Products Ltd
NEC Viewtechnology Ltd
Neostar Technology Co Ltd
Netflix
Netgem
NexGen Mediatech Inc
Ngai Lik Digital Technology Limited
Nian Yeong Enterprise Co Ltd
Nice Fountain Industrial Co Ltd
NienYi Industrial Corp
Niketech Electronic Corp
Nikon Corporation
Ningbo Bestwinning Technology Co Ltd
Ningbo Broad Telecommunication Co Ltd
Ningbo Longda Electronic Technology
NINGBO SHUNSHENG COMMUNICATION APPARATUS CO.,LTD
Nits Technology Inc.
Northstar System Corp
NOW Display & Innovation Co., Ltd.
NTK(HK)LIMITED-XINTANG FACTORY
Nucom Technology Corporation
Nvidia
Oak Technology, Inc.
Octal TV Technotrend Group
Ocvaco Holdings Ltd.
Omron Corporation
Onkyo Corporation
Opentech, Inc.
Ophit Co Ltd
Opticis Co Ltd
OQO Inc
Orcom International Pte Ltd

Orient Power Multimedia Ltd.
Orion Electric Co., Ltd.
Pace Micro Technology PLC
Packard Bell BV
Pak Heng Technology Shenzhen Co Ltd
Palit Microsystems HK Ltd.
Pan-International Precision Electronic Co, Ltd
Parade Technologies Inc
Paradigm Electronics Inc
Park Electronics Co Ltd
PC Partner Ltd
Pericom Semiconductor
Pine Technology Holdings Limited
Pioneer Corporation
Pirelli Broadband Solutions SpA
Pixel Magic Systems Ltd.
Plastron Precision Co Ltd
Point of View BV.
Poker Quattro Spa
Power Communication Tech Co Ltd
Premier Image Technology Corporation
Prime Electronics & Satellitics Inc
Proconn Technology Co. Ltd
Profilo Telra Elektronik San Ve Ticaret AS
Projection Design
Prolink Microsystems Corp
Promax Electronica SA
PROTON ELECTRONIC INDUSTRIAL CO LTD
Proview Group Limited
PT. Hit Electro Mechanics Indonesia
Pulse
Qingdao Haier Electronics Co Ltd
Qisheng International Ltd
Quanta Computer Inc.
Quantum Data Inc.
Race Point Media LLC
Radiient Technologies Inc
Radiospire Networks, Inc
Radix Inc
Rapid Conn Inc
REaltek Semiconductor Corp
Redmere Technology Ltd
Rikor TV Ltd
Rockwell Collins
Rohde & Schwarz GmbH & Co. KG
Rohm Co., Ltd.
Rotel Global Ltd
RTCom USA Inc
Runco International Inc
RZ Products Group LLC
S3 Graphics Co Ltd
Sampo Corporation
Samsung Electronics
Samsung Techwin Co Ltd
Samtec
SAMWIN HONG KONG LIMITED
Sandmartin Zhongshan Electronic Co Ltd
Santa Electronics, Inc.
Sanyo Electric Co., Ltd.
Sapphire Technology Ltd
Scientific-Atlanta, Inc.
Seiko Epson Corp.
Seongji Industrial Co Ltd
Sercomm Corporation
SHANGHAI FAR YEAR TECH CO.,LTD
Shanghai Haiying Electronics Co., Ltd.
Shanghai Yueshen Electronic Co Ltd
Sharp Corporation
Shenyang Tongfang Multimedia Co Ltd
Shenzhen ALEX Connector CO LTD
Shenzhen All-Best Digital Technology Co. Ltd.
Shenzhen Arlink Tech Co Ltd
Shenzhen Choseal industrial Co.,ltd
Shenzhen Coship Electronics Co Ltd
Shenzhen Delikang Electronics Technology Co., Ltd.
Shenzhen E-boda Technology Co
Shenzhen Founder Cyber Technology Co., Ltd
Shenzhen Giec Electronics Co Ltd
ShenZhen Junlan Electronic Ltd
Shenzhen KXD Multi-Media Co Ltd
Shenzhen Milkyway Electronic Ltd Corp
Shenzhen Oriental Digital Technology Co Ltd
Shenzhen Shunxun Electronics Co; Ltd.
ShenZhen Xing Feng Industry Co Ltd
Shenzhen Yongdefu Electronic Industry Co Ltd

Shibasoku Co Ltd
Shin Din Cable Ltd
Shin Kin Enterprises Co., Ltd.
Shinybow Technology Co Ltd
Shuttle Inc.
SHYH SHIUN TERMINALS Co., Ltd
Siauliu Tauro Televizoriai
Sichuan Changhong Electric Co Ltd
Sicon Semiconductor AB
Siemens Home & Office Communication GmbH & Co KG
Sigma Designs
Silicon Integrated Systems
Silicon Library Inc
Silicon Optix
SIM2 Multimedia S.p.A.
Sinai Technology Co Ltd
Singatron Enterprise Co Ltd.
Skyworth TV Holdings Ltd.
Smartasic Technology
SMK Corporation
Sonetek Technology Corp
Sotec Co., Ltd.
Space Shuttle (HK) Hi-Fi Wire & Cable Industry Co Ltd
Sparkle Computer Co Ltd
Spectrum Electronic Solutions Inc
Speed Conn Co. Ltd.
Speed Tech Corp
SRI Cable Technology Ltd
ST Microelectronics
Storm Electronics Co., Ltd
StreamTel Corporation
Sunf Pu Technology Co Ltd
Sunniwell Broadband Digital Science and Technology, Ltd
Sunplus Technology Co Ltd
Sure-Fire Electrical Corporation
Suyin Corporation
Suzhou Yuqiu Technology Co., Ltd
Suzhou Zhongxinglian Precision Industry Co Ltd
SVA Information Industry Co Ltd
SVP Corporate Strategy
Sweetray Industrial Ltd
Ta Yun Terminals Industrial Co., Ltd.
Tai Sol Electronics
Tai Twun Enterprise Co Ltd
Tai Yip Electrical Co,. Ltd
Taiko Denki Co., Ltd.
Taiwin Electronics Co; Ltd.
Taiyo Yuden Co Ltd
Tatung Co.
T-Conn Precision Corporation
TDK Corporation
Teac Corporation
Teams Power Limited
Tech Sources Co Ltd
Tech Technology Industies Company Limited
TechniSat Digital GmbH
Techview International Technology Inc
Tecnimagen S.A.
TECO Electric & Machinery Co Ltd
Tekcon Electronics Corp
Tektronix
Telsey S.p.A.
Texas Instruments Inc
Thintek Optronics Corp
Tiger Technology Inc.
Tikson Electronics Ltd
Tilgin AB
Time Interconnect Ltd.
TiVo Inc.
Tokyo Sinbon Electronics Co., Ltd.
Tonetron Electronic Ltd
Tonic Digital Products Ltd.
Top Powersonic Co Ltd
Top Victory Technology Limited
Topfield Co Ltd
Total Technologies Ltd
Trans Electric Co Ltd
Trident Microsystems Far East Ltd
Tsay-E International Inc (BVI)
TTE Corporation
Tul Corporation
TV One
Tyco Electronics Corporation
TZYR HWEY ENTERPRISE CO. LTD
Ubicod Co. Ltd.
UEC Technologies Ptv Ltd

Uniden Corporation
Unigraf Oy
Universal Abit Co Ltd
Unixtar Technology Inc.
Value Platforms Ltd
Vativ Technologies
VDC Technologies SPA
Vestel Elektronik Sanayi Ve Ticaret A.S.
Via Technologies Inc
Vicxon Corporation
Videon Central
View Net A/S
Viewsonic Corporation
Vinci Labs
VizionWare, Inc.
VSO Electric Co; Ltd.
VUDU, Inc.
Wafe Hardware Electronics Co. Ltd
Wan Lung Electric Wire & Cable
Wealleys Technologies Co Ltd
Wegener Communications Inc
WhenZhou Guang Ye Electronic Co Ltd
Wieson Technologies Co., Ltd.
Winbase Electronics Co Ltd
Wing-Span Enterprise Co., Ltd
Winpoint Electronic Corp.
Wireless Ventures
Wiretek International Investment Ltd
WIS Technologies Inc
Wistron Corporation
Woosung Nextier Corp
Worldwide Cable International Ltd
Wyplay SAS
Xiamen Overseas Chinese Electronic Co., Ltd.
Ya Hsin Industrial Co Ltd
Yamaha Corporation
YC Cable USA Inc
YEONG DER (SUM-EM)ENTERPRISES CO., LTD.
Yesin Electronics Technology Co., Ltd.
YFC-Boneagle Electric Co Ltd
YM Electric Co., Ltd.
Yokogawa Electric Corporation
Y-S Electronic Co., Ltd
YuXing Electronics Company Ltd
Zentek Technology Japan Inc
Zhejiang Tianle Digital Electric Co Ltd
Zhong Ming Wire Cable Technology Co Ltd
Zhong Shan Shi Neon Electronic Factory Ltd
Zhongshan DingCai AV Technology Co Ltd
Zhongshan Jointek Digital Technology Ltd
Zhongshan Tomei Audio & Video Products Co
Zhuhai Nintaus Electrical Appliances Co,.Ltd.
Zinwell Corporation
Ziova Corporation Pty Ltd
Zoran Corporation
ZyXel Communications Corp

©2006HDMI ALL RIGHTS RESERVED / TERMS POLICIES

https://web.archive.org/web/20070630082113/http://www.hdmi.org/manufacturer/specification.aspx

INTERNET ARCHIVE WayBackMachine
http://www.hdmi.org/manufacturer/specification.aspx
Go MAY JUN 30 2007 AUG 2006 2008
202 captures
16 Jun 2007 - 5 Sep 2
About this capture

HDMI
HIGH-DEFINITION MULTIMEDIA INTERFACE

Stay Connected
Sign up for updates from HDMI



Specification

The HDMI technical specification is available to HD device, display, component, and test equipment manufacturers at no charge. In order to download the specification, or become an HDMI Adopter, please complete the form that follows.

First Name:

Last Name:

Email:

Company:

Country: United States

Industry: Consumer Electronics Manufacturer

In your job function, which products are you involved with? (Please select all that apply.)

- ☐ Cables
- ☐ DTVs
- ☐ Standard Definition DVD Player/Recorders
- ☐ High Definition Optical Disc Players/Recorders (HD DVD & Blu-ray)
- ☐ A/V Receiver/Pre-Amplifier
- ☐ Projectors
- ☐ Camcorders and digital still cameras
- ☐ Mobile Phones
- ☐ Portable Media Players
- ☐ Game Consoles & Handheld Games
- ☐ Repeaters/Boosters
- ☐ Switches & Splitters
- ☐ "Home Theater in a Box" (Bundled AV receiver/DVD Player)
- ☐ PCs (Desktop/Notebooks)
- ☐ PC Video Cards
- ☐ PC Displays/Monitors
- ☐ Set Top Boxes/DVRs
- ☐ (Cable/Satellite/Terrestrial/IP/DSL)
- ☐ Other (Please specify)

Terms of Use: Please review the following terms of use in order to download the HDMI Specification solely for informational purposes:

```
HDMI SPECIFICATION
INFORMATIONAL AGREEMENT
IMPORTANT - READ CAREFULLY.

THIS AGREEMENT (THE "AGREEMENT") IS A LEGALLY
BINDING AGREEMENT BETWEEN YOU ON BEHALF OF
YOURSELF AND, IF YOU ARE USING THE SPECIFICATION
IN A PROFESSIONAL CAPACITY, YOUR EMPLOYER
(COLLECTIVELY "RECIPIENT" OR "YOU"), AND HDMI
```

○ **Agree** ◉ **Disagree**

Stay Connected: ☐ I would like to receive updates from HDMI Licensing, LLC.

By choosing to opt-in, you agree that your contact information may be used by HDMI Licensing, LLC for the purposes of sending you information on HDMI-related items.

HDMI respects your desire for privacy. Personal information will not be shared with organizations without your consent, except to process data on HDMI Licensing, LLC's behalf. Please refer to HDMI's privacy policy for more information.

Submit Cancel

©2006 HDMI. ALL RIGHTS RESERVED / TERMS POLICIES

https://web.archive.org/web/20090621022447/http://www.hdmi.org:80/manufacturer/specification.aspx

INTERNET ARCHIVE Wayback Machine http://www.hdmi.org/manufacturer/specification.aspx
202 captures 16 Jun 2007 - 5 Sep 2
INTERNET ARCHIVE Wayback Machine http://www.hdmi.org/manufacturer/specification.aspx Go APR JUN AUG
202 captures 16 Jun 2007 - 5 Sep 2 2008 21 2009 2010 About this capture

HDMI HIGH-DEFINITION MULTIMEDIA INTERFACE

简体中文 > Google™ Custom Search Adopter Sign In >

One Cable. One Standard. The future-ready way to connect HD.

Home | Manufacturers | Installers | Consumers | Resources | News & Events | Press

- Technology
- Introducing HDMI 1.4
- Specification
- Trademark & Logo Guidelines
- Becoming an Adopter
- Adopter Extranet
- Approved Connector List
- Testing Policies
- Authorized Test Centers

STAY CONNECTED
Sign up for updates from HDMI

Related Links

- Contact Us
- List of HDMI Adopters
- Knowledge Base
- FAQ
- Logos & Photos

Manufacturers



home > manufacturers > specification

Specification

On Monday, June 8, 2009, HDMI Licensing, LLC issued a press release announcing the release of the HDMI Specification Ver.1.4, now available to Adopters. This latest version of the HDMI specification will, among other advancements, offer networking capabilities with Ethernet connectivity and add an Audio Return Channel to enable upstream audio connections via the HDMI cable.

The HDMI Specification Ver.1.4 is now available for download by Adopters.

For those who wish to download the HDMI Specification Ver.1.3a, please click here.

© 2003-2009 HDMI Licensing, LLC. All rights reserved.

About Us / Terms / Policies / Contact Us

https://web.archive.org/web/20061129023219/http://www.hdmi.org/about/adopters_founders.asp

INTERNET ARCHIVE WayBackMachine
http://www.hdmi.org/about/adopters_founders.asp
Go
SEP NOV JAN
29
2005 2006 2007
143 captures
14 Nov 2004 - 16 Jun
About this capture

MANUFACTURER
LOGOS & PHOTOS
LEARNING CENTER
ABOUT
Adopters & Founders
FAQ
PRESS
CONTACT



Adopters & Founders

HDMI Founders

Hitachi, Ltd.
Matsushita Electric Industrial Co., Ltd. (Panasonic)
Philips Consumer Electronics International B.V.
Silicon Image, Inc.
Sony Corporation
Thomson, Inc.
Toshiba Corporation

HDMI Adopters

Last Updated on 11/27/2006

3S Digital Co Ltd
A&R Cambridge Ltd (Arcam)
Across Techno Inc
Advance & Beyond-Tech Co Ltd
Advanced-Connectek Inc.
Advanced Digital Broadcast Polska Sp zoo
Advanced Display Lab Inc.
Advanced Micro Devices
Agilent Technologies Inc.
Aiter Cable (Qinfen Aoya)
AJA Video Systems, Inc.
Akai Electric Co Ltd
Alco Digital Devices Limited
Algolith Inc
ALI Corporation
Allion Computer Inc
Alltop Technology Co Ltd
Amino Communications Ltd
Amoi Electronics Co Ltd
Amphenol Corporation
Amstrad PLC
AmTran Technology
AMX
Analog Devices, Inc.
Analogix Semiconductor Inc
Anam Electronics Co Ltd
Anchor Bay Technologies
Antler Computer BV
Apple Computer
ARC Electro Industry Corp.
Arion Technology Inc
Asahi Kasei Microsystems Co Ltd
AstroDesign Inc.
Asustek Computer Inc
Atec Co Ltd
Aten International Co Ltd
Audio Partnership PLC
Audio-Technica Corporation
Audiovox Electronics Corporation
Aurora Multimedia Corp
Avid Technology Inc.
Avision Technology (Changzhou) Co., Ltd
Avocent Corporation
BAFO Technologies/TWN 1st Line Elect Cor
Bang & Olufsen
BBK AV Electronics Corp Ltd.
BenQ Corporation
Beko Elektronik A.S.
Beyondwiz Co Ltd
BizLink Technology
Blackmagic Design
Bose Corporation
Broadcom Corporation
C&C Technic Taiwan Co Ltd
Cable Vision Electronics Co Ltd
Calibre UK LTD
California Micro Devices
Cameo Communications Inc
Canon Inc
Carol Wiring Harness Co Ltd
Cary Audio Design Ltd
Celrun Co Ltd
Changzhou Wujin Best Electronic Cables Co
Changzhou Jinyin Futao Photoelectronic Devices Co
Chaoyue Jiangsu Digital Co Ltd
Cheertek Inc
Krell Industries
KTV Global Corp
KunZhong Electronic Co Ltd
Leader Electronics Corp.
Leader Wise Technology HK Ltd
Leadtek Research Inc
LeCroy Corporation
Le Hong PO Company Ltd
Leoco Corporation
LG Electronics
Link Concept Technology Ltd
Linkiss Co Ltd
Linktec Technologies Co Ltd
Linn Products Ltd
Lin Shiung Enterprise Co Ltd
Loewe Opta GmbH
Logitech Inc
Longwell Company
Lorom Industrial Co Ltd
LSI Logic Corporation
Luxshare Electronical Ltd
Macro Image Technology Inc
Magnum Semiconductor
Main Super Enterprises Co.
Marusys Co Ltd
Marshall Electronics
Marvell International Ltd
Master Co. LTD-Korea
Master Hill Electric Wire & Cable
Maxim Integrated Products
MediaTek Corp.
Meiloon Industrial Co., Ltd.
Meridian Audio Limited
Metz-Werke GmbH & Co KG
Micro-Star International Co Ltd
Micronas GmbH
Microsoft Corporation
Microtek International Inc
MIK21 Co., Ltd
Y-S Electronic Co., Ltd
Mitsubishi Electric Corporation
Mitsumi Electric Co., Ltd.
Mivar Di Carlo Vichi Sapa
Molex Incorporated
Morning Star Industrial
Motorola, Inc.
MovieBeam Inc
MSL Enterprises Corp
Mstar Semiconductor, Inc.
Murata Manufacturing Co.,Ltd.
Musco Sports Lighting LLC
NAD Electronics International
Nanjing Wanlida Technology Co Ltd
NEC Display Solutions Inc
NEC Electronics Corporation
NEC Personal Products Ltd
NEC Viewtechnology Ltd
Neostar Technology Co Ltd
Netflix
Netgem
NexGen Mediatech Inc
Nian Yeong Enterprise Co Ltd
Nice Fountain Industrial Co Ltd
Nikon Corporation
NienYi Industrial Corp
Niketech Electronic Corp

Cheertek Inc
Cheng Uei Precision Industry Co Ltd
Chi Mei Optoelectronics Corp
Chi Shing Computer Accessories Co Ltd
Ching Tai Electric Wire & Cable Co Ltd
Chip Advanced Technology
Chroma ATE, Inc.
Chrontel Inc
Cinemateq GmbH & Co KG
Cinetron Technology Inc.
Circuit Assembly Corporation
Cisco Systems Inc-Linksys
Comax Technology Inc.
Comda Advanced Technology Corp
Comoss Electronic Co., Ltd.
Compal Electronics Inc
Conexant Systems, Inc.
Connexions Technology Corp
Conrac GmbH
Contac Cable Industrial Ltd
Contek Electronics Co Ltd
Copartner Technology Co Ltd
Coretronic Corporation
CORPO-RIGHT ENTERPRISE CO., LTD
Cosmic Digital Technology Co Ltd
Coulomb Electronics Ltd
Crestron Electronics
CVT International Ltd
Cypress Semiconductor
Cypress Technology Co Ltd
D&M Holdings, Inc.
DA2 Technologies Corporation
Daewoo Electronics Corporation
Dalian Golden Hualu Digital Technology Co Ltd
David Electronics Co Ltd
Delta Electronics Inc.
Der An Electric Wire & Cable Co Ltd
Desay A&V Science And Technology Co., Ltd
DiBoss Co Ltd
Digeo Interactive LLC
Digimedia Technology Co Ltd
Digital Device Inc
Digital Stream Technology Inc.
Digitall World Co Ltd
DigitalZone Co Ltd
D-Link Corporation
DM Technology Co Ltd
Dongguan Alllike Electronics Co Ltd
Dongguan Haurtian Wire & Cable Co Ltd
Dongguan Qisheng Electronic Ind
Dongguan Shinko Wire Co Ltd
Dongguan Teconn Connector Ltd
Dongguan Xinya Electronic Technology Co Ltd
Dtrovision LLC
Eastern Asia Technology Ltd.
Echostar Technologies Corp
Efficere Technologies
Eizo Nanao Corporation
Elka International Ltd
Ellion Digital Inc
Emine Technology Co Ltd
Energy Master Co Ltd
Enseo Inc
Enuclia Semiconductor
Erae Electronics
ESS Technology, Inc.
Esto Inc
Eternal Fine Technology Development Ltd
Etronics Corporation
Ever Connect Technology Co Ltd
Ever Great Electric Wire and Cable Co Ltd
Ever Much Company Ltd
Evernew Wire & Cable Co Ltd
EVGA Corporation
Explore Microelectronics, Inc.
Extraordinary Technologies Ptd Ltd (Halcro)
Fairchild Semiconductor
FCI USA Inc.
Foosung Tech Corp
Formosa Electric Wire & Cable
Foxconn Electronics Inc.
Freebox
Freeport Resources Enterprises Corp
FTD Solutions Pte., Ltd.
Fuji Parts Co Ltd
Fujitsu General Limited
Niketech Electronic Corp
Ningbo Bestwinning Technology Co Ltd
Ningbo Broad Telecommunication Co Ltd
NINGBO SHUNSHENG
Nucom Technology Corporation
COMMUNICATION APPARATUS CO.,LTD
Northstar System Corp
NTK(HK)LIMITED-XINTANG FACTORY
Nvidia
Oak Technology, Inc.
Octal TV Technotrend Group
Onkyo Corporation
Ophit Co Ltd
Opticis Co Ltd
OQO Inc
Orcom International Pte Ltd
Orient Power Multimedia Ltd.
Orion Electric Co., Ltd.
Pace Micro Technology PLC
Packard Bell BV
Pak Heng Technology Shenzhen Co Ltd
Parade Technologies Inc
Paradigm Electronics Inc
Park Electronics Co Ltd
PC Partner Ltd
Pericom Semiconductor
Pioneer Corporation
Pirelli Broadband Solutions SpA
Pixel Magic Systems Ltd.
Plastron Precision Co Ltd
Poker Quattro Spa
Power Communication Tech Co Ltd
Premier Image Technology Corporation
Prime Electronics & Satellitics Inc
Profilo Telra Elektronik San Ve Ticaret AS
Prolink Microsystems Corp
Promax Electronica SA
Proview Group Limited
PT. Hit Electro Mechanics Indonesia
Pulse
Qingdao Haier Electronics Co Ltd
Qisheng International Ltd
Quantum Data Inc.
Race Point Media LLC
Radix Inc
Radiient Technologies Inc
Rapid Conn Inc
REaltek Semiconductor Corp
Redmere Technology Ltd
Rockwell Collins
Rohm Co., Ltd.
Rotel Global Ltd
RTCom USA Inc
Runco International Inc
RZ Products Group LLC
S3 Graphics Co Ltd
Sampo Corporation
Samsung Electronics
Samsung Techwin Co Ltd
Samtec
Sandmartin Zhongshan Electronic Co Ltd
Santa Electronics, Inc.
Sanyo Electric Co., Ltd.
Sapphire Technology Ltd
Scientific-Atlanta, Inc.
Seiko Epson Corp.
Seongji Industrial Co Ltd
Sercomm Corporation
Sharp Corporation
Shenyang Tongfang Multimedia Co Ltd
Shenzhen Arlink Tech Co Ltd
Shenzhen Coship Electronics Co Ltd
Shenzhen E-boda Technology Co
Shenzhen Giec Electronics Co Ltd
ShenZhen Junlan Electronic Ltd
Shenzhen KXD Multi-Media Co Ltd
Shenzhen Milkyway Electronic Ltd Corp
Shenzhen Oriental Digital Technology Co Ltd
ShenZhen Xing Feng Industry Co Ltd
Shenzhen Yongdefu Electronic Industry Co Ltd
Shibasoku Co Ltd
Shin Din Cable Ltd
Shinybow Technology Co Ltd
Siauliu Tauro Televizoriai
Sichuan Changhong Electric Co Ltd
Siemens Home & Office Communication GmbH &

Fujitsu Limited
Fujitsu Siemens Computers GmbH
Funai Electric Co Ltd
Galaxy Technology (HK) Ltd.
Gateway Companies Inc
Gearway Electronics Dong Guan Co Ltd
Gefen, Inc.
Genesis Components Inc
Genesis Microchip Inc
Gericom AG
Genix InfoComm Co Ltd
Gennum Corporation
Gigabyte Technology Co Ltd
Global Technologies Inc
Glory Mark Electronic Ltd.
Golana Technology Corporation
Gold Rain Enterprises Corp
Golden Bridge Electech Inc.
Goodwang International Company
Gowell Electronic Ltd
GP Electronics SZ Limited
GPNC Co Ltd
Grandbeing Technology Co Ltd
Greatland Electronics Taiwan Ltd
Handan BroadInfoCom Co Ltd
Hannspree, Inc.
Harman International Industries Inc.
Harsper Co Ltd
Hightech Information System Ltd
High Top Precision Electronic Co. Ltd
Himax Technologies Inc.
Hip Hing Cable & Plug Mfy Ltd
Hirakawa Hewtech
Hirose Electric Co Ltd.
Hisense Electric Corp Ltd
Homecast Co Ltd
HongLin Electronic Co Ltd
Horng Tong Enterprise Co Ltd
Hosiden Corporation
HPC Technology Inc
H.P.R. Electronics GmbH
Huipu Electronic Shenzhen Co Ltd
Humax Co Ltd
Hyosung Electronics Co Ltd
Hyundai ImageQuest Co Ltd
i3 Micro Technology ab
I-O Data Device Inc
Idar Electronics Co Ltd
IDK Corporation
Imagenics Co Ltd
Indtek Co Ltd
Info-Tek Corporation
InFocus Corporation
Innovision Multimedia Ltd
Integrated Device Technology Inc
Intel Corporation
Intersil Corporation
Inventec Corporation
Jabil Circuit Inc
Jalco Co., Ltd.
Jalink Technology Inc.
Japan Aviation Electronics (JAE)
Jebsee Electronics Co Ltd
Jenving Technology AB / Supra Cables
Jean Co Ltd
Jesmay Electronics Co Ltd
Jhen Vei Electronic Co Ltd
Jiangsu Shinco Electronic Group Co Ltd
JIB Enterprise Group Co Ltd
Ji-Haw Industrial Co Ltd
Joinsoon Electronics MFG, Co LTD
Jye Tai Precision Industrial Co Ltd.
Kaleidescape, Inc.
Kawasaki Microelectronics Inc
KCS Co Ltd
Kechenda Plastic & Electronic Factory
KEL Corporation
Kenmark Industrial Co Ltd
Key Digital Systems
Konka Group Co Ltd
Kordz Pty Ltd
Kramer Electronics Ltd
Co KG
Sigma Designs
Silicon Library Inc
Silicon Optix
SIM2 Multimedia S.p.A.
Sinai Technology Co Ltd
Singatron Enterprise Co Ltd.
Skyworth TV Holdings Ltd.
Smartasic Technology
SMK Corporation
Sonetek Technology Corp
Space Shuttle (HK) Hi-Fi Wire & Cable Industry Co Ltd
Sparkle Computer Co Ltd
Spectrum Electronic Solutions Inc
SRI Cable Technology Ltd
ST Microelectronics
Sunf Pu Technology Co Ltd
Sunplus Technology Co Ltd
Sure-Fire Electrical Corporation
Suyin Corporation
Suzhou Yugiu Technology Co Ltd
Suzhou Zhongxinglian Precision Industry Co Ltd
Sweetray Industrial Ltd
T-Conn Precision Corporation
Ta Yun Terminals Industrial Co., Ltd.
Tai Sol Electronics
Tai Twun Enterprise Co Ltd
Taiko Denki Co., Ltd.
Taiyo Yuden Co Ltd
Tatung Co
TDK Corporation
Teac Corporation
Teams Power Limited
Tech Sources Co Ltd
TechniSat Digital GmbH
Tecnimagen S.A.
TECO Electric & Machinery Co Ltd
Techview International Technology Inc
Tekcon Electronics Corp
Tektronix
Texas Instruments Inc
Thintek Optronics Corp
Tikson Electronics Ltd
Tilgin AB
Time Interconnect Ltd.
TiVo Inc.
Tokyo Sinbon Electronics Co., Ltd.
Tonetron Electronic Ltd
Tonic Digital Products Ltd.
Topfield Co Ltd
Top Powersonic Co Ltd
Top Victory Technology Limited
Trans Electric Co Ltd
Trident Microsystems Far East Ltd
Tsay-E International Inc (BVI)
TTE Corporation
Total Technologies Ltd
Tul Corporation
TV One
Tyco Electronics Corporation
UEC Technologies Pty Ltd
Uniden Corporation
Universal Abit Co Ltd
Unixtar Technology Inc.
Vativ Technologies
Vestel Elektronik Sanayi Ve Ticaret A.S.
Via Technologies Inc
Videon Central Inc.
View Net A/S
Viewsonic Corporation
Vinci Labs
Wan Lung Electric Wire & Cable
Wealleys Technologies Co Ltd
Wegener Communications Inc
WhenZhou Guang Ye Electronic Co Ltd
Wieson Technologies Co., Ltd.
Winbase Electronics Co Ltd
Wiretek International Investment Ltd
WIS Technologies Inc
Wistron Corporation
Woosung Nextier Corp
Worldwide Cable International Ltd
X-Emi Inc
Xiamen Overseas Chinese Electronic
XN Technologies Inc

XN Technologies Inc
Ya Hsin Industrial Co Ltd
Yamaha Corporation
YC Cable USA Inc
YFC-Boneagle Electric Co Ltd
Yokogawa Electric Corporation
YuXing Electronics Company Ltd
Zentek Technology Japan Inc
Zhong Ming Wire Cable Technology Co Ltd
Zhongshan DingCai AV Technology Co Ltd
Zhongshan Jointek Digital Technology Ltd
Zhongshan Tomei Audio & Video Products Co
Zhong Shan Shi Neon Electronic Factory Ltd
Zhuhai Nintaus Electronic Industry Co Ltd
Zinwell Corporation
Ziova Corporation Pty Ltd
Zoran Corporation
ZyXel Communications Corp

©2006 HDMI. ALL RIGHTS RESERVED / Terms of Use - Privacy Policy

https://web.archive.org/web/20061211204205/http://www.hdmi.org:80/manufacturer/specification.asp

INTERNET ARCHIVE WayBackMachine
http://www.hdmi.org/manufacturer/specification.asp
Go
NOV 2005 | DEC 11 2006 | JAN 2008
74 captures
14 Nov 2004 - 5 May
About this capture

MANUFACTURER
Technology
Specification
Trademark & Logo
Becoming an Adopter
Adopter Extranet
Approved Connector List
Testing Policies
Authorized Test Centers
LOGOS & PHOTOS
LEARNING CENTER
ABOUT
PRESS
CONTACT

Specification

The HDMI technical specification is available to HD device, display, component, and test equipment manufacturers at no charge. In order to download the specification, or become an HDMI Adopter, please complete the form that follows.

1. Last Name:
2. First Name:
3. Title:
4. Phone:
5. Email:
6. Job Function:
   - ○ Director
   - ○ Engineer
   - ○ General Council/Legal
   - ○ Legal
   - ○ Manager
   - ○ Marketing
   - ○ Product Development
   - ○ Sales
   - ○ Vice President/Executive
   - ○ Other
7. Company:
8. Company Address:
   City:
   Country: United States
9. Type of Company: (check all that apply)
   - ☐ Consumer Electronics Manufacturer
   - ☐ PC Manufacturer
   - ☐ Display Manufacturer
   - ☐ Semiconducter Manufacturer
   - ☐ System Operator (Satellite, Cable)
   - ☐ Test Equipment Maker
   - ☐ Content Provider
   - ☐ Broadcaster
   - ☐ Trade Association
   - ☐ Other

Submit



©2006 HDMI. ALL RIGHTS RESERVED / Terms of Use - Privacy Policy

https://web.archive.org/web/20051125014531/http://standards.ieee.org/getieee802/download/802.3af-2003.pdf

INTERNET ARCHIVE
WayBackMachine
http://standards.ieee.org/getieee802/download/802.3af-2003.pdf
Go
50 captures
25 Nov 2005 - 10 Dec
OCT NOV FEB
25
2004 2005 2008
About this capture

Get IEEE 802® Program

About the Program
How to Support *or* Sponsor the Program
LAN/MAN Standards Subscription
IEEE 802® Working Groups
IEEE 802® Draft Standards
Wireless Zones
PDF Download Help

Get IEEE 802®

...Partnering with industry to close the digital divide

Terms of Use for IEEE Standards Publications Delivered in Electronic Form

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC., SHALL DELIVER THIS IEEE STANDARDS PUBLICATION TO YOU IN, ELECTRONIC FORMAT, ONLY ON THE CONDITION THAT YOU ACCEPT THE FOLLOWING TERMS OF USE.

READ THE TERMS OF USE CAREFULLY. BY PROVIDING THE REQUIRED INFORMATION AND SELECTING THE "ACCEPT" BUTTON AT THE BOTTOM OF THIS PAGE, YOU AGREE TO BE LEGALLY BOUND BY THESE TERMS OF USE CONTAINED HEREIN. IF YOU SELECT "DO NOT ACCEPT", THE FILE TRANSFER FOR THIS STANDARDS PUBLICATION WILL NOT CONTINUE.

COPYRIGHT

This IEEE Standards Publication ("Document") is approved by the IEEE Standards Association ("IEEE-SA") Standards Board and is published in accordance with established IEEE-SA Standards Board bylaws and operations procedures.

The Institute of Electrical and Electronics Engineers, Inc., ("IEEE") owns the copyright to this Document in all forms of media. Copyright in the text retrieved, displayed or output from this Document is owned by IEEE and is protected by the copyright laws of the United States and by international treaties. The IEEE reserves all its rights not expressly granted herein.

The IEEE is providing the Document to you at no charge. However, ***the Document is not to be considered "Public Domain"***, as the IEEE is, and at all times shall remain, the sole copyright holder in the Document.

TERMS OF USE

You may retrieve, download and print one (1) copy of this Document for your personal use. You may retain one (1) additional copy of this Document as your personal archive copy.

Except as allowed by the copyright laws of the United States of America or applicable international treaties, or as explicitly allowed in these Terms of Use, you may not further copy, prepare, and/or distribute copies of this Document, nor significant portions of the document, in any form, without prior written permission from the IEEE.

Requests for permission to reprint this Document, in whole or in part, or requests for a license to reproduce and/or distribute this Document, in any form, must be submitted via email to **Standards Licensing and Contracts**, or in writing at:

IEEE Standards Association
445 Hoes Lane, P.O. Box, 1331
Piscataway, NJ 08855-1331, USA.

LIMITED WARRANTIES & LIMITATIONS OF REMEDIES

LICENSOR Does not warrant or represent the accuracy or content of the document and Expressly Disclaims any Express or Implied Warranty, including any Implied Warranty of Merchantability or Fitness for a Specific Purpose or that the use of the document is free from patent infringement. The document is supplied ONLY "AS IS."

TO REPORT TECHNICAL PROBLEMS OR TO MAKE INQUIRIES REGARDING ELECTRONIC DELIVERY OF THIS IEEE STANDARDS PUBLICATION:

If you are unable to receive this Document as requested, or if there is a problem with the transmission, you should contact **Product Support** who will make reasonable efforts to remedy the problem.

FOR INQUIRIES ABOUT THESE TERMS OF USE

If you have questions about these Terms of Use or wish to inquire about options for licensing this Document or any other IEEE Standards Publication, contact the **Standards Licensing and Contracts** for more information.

I acknowledge that I have read and fully understand the foregoing information and agree to abide by these Terms of Use.

USER TYPE *(Required)*

Please select one

ACCEPT/BEGIN DOWNLOAD

DO NOT ACCEPT

Copyright ©2004 IEEE-SA
(product-support@ieee.org)
URL: http://standards.ieee.org/getieee802/
(Modified: 8 Nov 04)

EXHIBIT B

File 001 - https://web.archive.org/web/20030411130221/http://www.serialata.org/collateral/zipdownloads/serialata10a.ZIP



Serial ATA:

High Speed Serialized AT Attachment

Revision 1.0a
7-January-2003

APT Technologies, Inc.
Dell Computer Corporation
Intel Corporation
Maxtor Corporation
Seagate Technology

This 1.0a revision of the Serial ATA / High Speed Serialized AT Attachment specification consists of the 1.0 revision of the specification with the following errata incorporated:

2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 29, 30, 31, 32, 33, 34, 35, 36, 38, 40

The following errata were withdrawn and were not incorporated into this revision of the specification:

1, 27, 28

Details on individual errata can be downloaded from the Serial ATA Working Group website at www.serialata.org.

This 1.0a revision of the Serial ATA / High Speed Serialized AT Attachment specification ("Final Specification") is available for product design. Product implementations should ensure compliance with this specification.

SPECIFICATION DISCLAIMER

THIS SPECIFICATION IS PROVIDED TO YOU "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE. THE AUTHORS OF THIS SPECIFICATION DISCLAIM ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY PROPRIETARY RIGHTS, RELATING TO USE OR IMPLEMENTATION OF INFORMATION IN THIS SPECIFICATION. THE AUTHORS DO NOT WARRANT OR REPRESENT THAT SUCH USE WILL NOT INFRINGE SUCH RIGHTS. THE PROVISION OF THIS SPECIFICATION TO YOU DOES NOT PROVIDE YOU WITH ANY LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS.

Copyright 2000, 2001, 2002, 2003, APT Technologies, Inc., Dell Computer Corporation, Intel Corporation, Maxtor Corporation, Seagate Technology LLC. All rights reserved.

For more information about Serial ATA, refer to the Serial ATA Working Group website at www.serialata.org.

All product names are trademarks, registered trademarks, or servicemarks of their respective owners.

Serial ATA Workgroup Technical Editor:

Klaus-Peter Deyring
APT Technologies, Inc.
1347 Pacific Avenue
Suite 205
Santa Cruz, CA 95060 USA
Tel: 831-429-7262
Fax: 831-429-7272
Email: pete@apt-tech.com

Serial ATA Workgroup Promoters



Klaus-Peter Deyring
(831) 429-7262
pete@apt-tech.com

APT Technologies, Inc.
1347 Pacific Avenue
Suite 205
Santa Cruz, CA 95060



Tom Pratt
(512) 723-8549
tom_pratt@dell.com

Dell Computer Corp.
One Dell Way
Round Rock, TX 78682



Knut Grimsrud
(503) 264-8419
knut.s.grimsrud@intel.com

Intel Corp.
M/S JF2-53
2111 NE 25th Ave
Hillsboro, OR 97124



Farbod Falakfarsa
(408) 894-4066
Farbod.Falakfarsa@maxtor.com

PTD Architecture Group
500 McCarthy Blvd
Milpitas, CA 95035



Marc Noblitt
I/O Planning manager
(720) 684-1333 phone
(720) 684-1031 fax
marc_noblitt@seagate.com

Seagate Technology
389 Disc Drive
Longmont, CO 80503

TABLE OF CONTENTS

1 Scope ... 11
2 Goals, objectives and migration considerations ... 11
2.1 Goals and objectives ... 11
2.2 Migration Considerations ... 12
3 Normative references ... 13
3.1 Approved references ... 13
3.2 Other references ... 13
4 Definitions, abbreviations, and conventions ... 13
4.1 Definitions and abbreviations ... 13
4.1.1 ATA (AT Attachment) ... 13
4.1.2 ATAPI (AT Attachment Packet Interface) device ... 13
4.1.3 backchannel ... 14
4.1.4 bit synchronization ... 14
4.1.5 byte ... 14
4.1.6 character ... 14
4.1.7 character alignment ... 14
4.1.8 character slipping ... 14
4.1.9 code violation ... 14
4.1.10 comma character ... 14
4.1.11 comma sequence ... 14
4.1.12 command aborted ... 15
4.1.13 command completion ... 15
4.1.14 command packet ... 15
4.1.15 Control Block registers ... 15
4.1.16 control character ... 15
4.1.17 control variable ... 15
4.1.18 CRC ... 15
4.1.19 data character ... 15
4.1.20 device ... 15
4.1.21 DMA (direct memory access) ... 15
4.1.22 Dword ... 16
4.1.23 Dword synchronization ... 16
4.1.24 encoded character ... 16
4.1.25 elasticity buffer ... 16
4.1.26 First-party DMA access ... 16
4.1.27 First-party DMA mode ... 16
4.1.28 FIS ... 16
4.1.29 Frame Information Structure ... 16
4.1.30 frame ... 16
4.1.31 interrupt pending ... 17
4.1.32 legacy device ... 17
4.1.33 legacy mode ... 17
4.1.34 legal character ... 17
4.1.35 LFSR ... 17
4.1.36 PIO (programmed input/output) ... 17
4.1.37 primitive ... 17
4.1.38 sector ... 17
4.1.39 Shadow Register Block registers ... 17
4.1.40 unrecoverable error ... 17
4.1.41 word ... 17
4.2 Conventions ... 18
4.2.1 Precedence ... 18
4.2.2 Keywords ... 18

4.2.3 Numbering 19
4.2.4 Signal conventions 19
4.2.5 Bit conventions 19
4.2.6 State diagram conventions 20
4.2.7 Timing conventions 21
4.2.8 Byte, word and Dword Relationships 22
5 General overview 23
5.1 Sub-module operation 25
5.2 Standard ATA Emulation 26
5.2.1 Software Reset 26
5.2.2 Master-only emulation 26
5.2.3 Master/Slave emulation (optional) 27
5.2.4 Standard ATA interoperability state diagrams 29
5.2.5 IDENTIFY DEVICE command 33
6 Physical layer 36
6.1 Overview 36
6.2 List of services 36
6.3 Cables and connectors specifications 37
6.3.1 Overview 37
6.3.2 Objectives 37
6.3.3 General descriptions 37
6.3.4 Connector configurations and locations 40
6.3.5 Mating interfaces 43
6.3.6 Serial ATA cable 56
6.3.7 Backplane connector configuration and blind-mating tolerance 56
6.3.8 Connector labeling 57
6.3.9 Connector and cable assembly requirements and test procedures 57
6.4 Low level electronics block diagram 68
6.4.1 Diagram 68
6.4.2 Physical plant overall block diagram description 69
6.4.3 Analog front end (AFE) block diagram description 72
6.5 General specifications 73
6.5.1 System 73
6.6 Module specifications 74
6.6.1 Definitions 74
6.6.2 Electrical specifications 74
6.6.3 Differential voltage/timing (EYE) diagram 78
6.6.4 Sampling jitter specifications 79
6.7 Functional specifications 86
6.7.1 Overview 86
6.7.2 Common-mode biasing 86
6.7.3 Matching 88
6.7.4 Out of band signaling 88
6.7.5 Idle bus condition 97
6.7.6 Elasticity buffer management 97
6.7.7 Test considerations 98
6.8 Interface power states 117
6.8.1 Interface power state sequences 118
7 Link layer 129
7.1.1 Overview 129
7.2 Encoding method 129
7.2.1 Notation and conventions 129
7.2.2 Character code 131
7.2.3 Transmission summary 139
7.2.4 Reception summary 141
7.3 Transmission overview 142

7.4 Primitives 145
7.4.1 Overview 145
7.4.2 Primitive descriptions 145
7.4.3 Primitive encoding 148
7.4.4 Abort primitive 148
7.4.5 Continue primitive 149
7.4.6 ALIGN primitive 152
7.4.7 Flow control signaling latency 152
7.4.8 Examples 156
7.5 CRC calculation and scrambling of FIS contents 159
7.5.1 CRC 159
7.6 Link layer state diagrams 161
8 Transport layer 184
8.1 Transport layer overview 184
8.1.1 FIS construction 184
8.1.2 FIS decomposition 184
8.2 Frame Information Structure (FIS) 184
8.3 Overview 184
8.4 Payload content 185
8.5 FIS types 185
8.5.1 All FIS types 185
8.5.2 Register - Host to Device 186
8.5.3 Register - Device to Host 188
8.5.4 Set Device Bits - Device to Host 190
8.5.5 DMA Activate - Device to Host 191
8.5.6 DMA Setup – Device to Host or Host to Device (Bidirectional) 192
8.5.7 BIST Activate - Bidirectional 194
8.5.8 PIO Setup – Device to Host 197
8.5.9 Data - Host to Device or Device to Host (Bidirectional) 199
8.6 Host transport states 201
8.6.1 Host transport idle state diagram 201
8.6.2 Host Transport transmit command FIS diagram 204
8.6.3 Host Transport transmit control FIS diagram 206
8.6.4 Host Transport transmit First-party DMA Setup – Device to Host or Host to Device FIS state diagram 207
8.6.5 Host Transport transmit BIST Activate FIS 209
8.6.6 Host Transport decompose Register FIS diagram 210
8.6.7 Host Transport decompose a Set Device Bits FIS state diagram 211
8.6.8 Host Transport decompose a DMA Activate FIS diagram 212
8.6.9 Host Transport decompose a PIO Setup FIS state diagram 215
8.6.10 Host Transport decompose a First-party DMA Setup FIS state diagram 219
8.6.11 Host transport decompose a BIST Activate FIS state diagram 220
8.7 Device transport states 221
8.7.1 Device transport idle state diagram 221
8.7.2 Device Transport send Register – Device to Host state diagram 223
8.7.3 Device Transport send Set Device Bits FIS state diagram 224
8.7.4 Device Transport transmit PIO Setup – Device to Host FIS state diagram 225
8.7.5 Device Transport transmit Legacy DMA Activate FIS state diagram 226
8.7.6 Device Transport transmit First-party DMA Setup – Device to Host FIS state diagram 227
8.7.7 Device Transport transmit Data – Device to Host FIS diagram 228
8.7.8 Device Transport transmit BIST Activate FIS diagram 230
8.7.9 Device Transport decompose Register – Host to Device state diagram 231
8.7.10 Device Transport decompose Data (Host to Device) FIS state diagram 232
8.7.11 Device Transport decompose DMA Setup – Host to Device or Device to Host state diagram 234

8.7.12 Device Transport decompose a BIST Activate FIS state diagram 234
9 Device command layer protocol 236
9.1 Power-on and COMRESET protocol diagram 236
9.2 Device Idle protocol 238
9.3 Software reset protocol 243
9.4 EXECUTE DEVICE DIAGNOSTIC command protocol 245
9.5 DEVICE RESET command protocol 247
9.6 Non-data command protocol 248
9.7 PIO data-in command protocol 250
9.8 PIO data-out command protocol 252
9.9 DMA data in command protocol 253
9.10 DMA data out command protocol 254
9.11 PACKET protocol 256
9.12 READ DMA QUEUED command protocol 261
9.13 WRITE DMA QUEUED command protocol 263
10 Host adapter register interface 266
10.1 SStatus, SError and SControl registers 267
10.1.1 SStatus register 268
10.1.2 SError register 269
10.1.3 SControl register 271
11 Error handling 272
11.1 Architecture 272
11.2 Phy error handling overview 273
11.2.1 Error detection 273
11.2.2 Error control actions 274
11.2.3 Error reporting 275
11.3 Link error handling overview 275
11.3.1 Error detection 275
11.3.2 Error control actions 275
11.3.3 Error reporting 276
11.4 Transport error handling overview 276
11.4.1 Error detection 277
11.4.2 Error control actions 277
11.4.3 Error reporting 278
11.5 Software error handling overview 279
11.5.1 Error detection 279
11.5.2 Error control actions 279
Appendix A. Sample Code for CRC and Scrambling 282
A.1 CRC calculation 282
A.1.1 Overview 282
A.1.2 Maximum frame size 282
A.1.3 Example code for CRC algorithm 282
A.1.4 Example code for CRC algorithm 282
A.1.5 Example CRC implementation output 284
A.2 Scrambling calculation 285
A.2.1 Overview 285
A.2.2 Example code for scrambling algorithm 285
A.2.3 Example scrambler implementation 285
A.2.4 Example scrambler implementation output 288
A.3 Example frame 289
Appendix B. Type field value selection 290
B.1 Type field values 290
Appendix C. Further information about cable and connector 292
C.1 Device connector configurations 292
C.1.1 Configuration 35A1 292
C.1.2 Configuration 35B1 292

Configuration 35B2 292
C.1.3 Configuration 35B3 293
C.1.4 Configuration 35B4 293
C.1.5 Configuration 35B5 293
C.1.6 Configuration 35C1 294
C.1.7 Configuration 25A1 294
C.2 Cable construction example 294
C.3 Contact material and plating 295
Appendix D. Command processing overview 297
D.1 Non-data commands 297
D.1.1 Legacy DMA read by host from device 298
D.1.2 Legacy DMA write by host to device 298
D.1.3 PIO data read from the device 299
D.1.4 PIO data write to the device 299
D.1.5 Queued DMA read from device 301
D.1.6 Queued DMA write to device 301
D.1.7 ATAPI Packet commands with PIO data in 302
D.1.8 ATAPI Packet commands with PIO data out 303
D.1.9 ATAPI Packet commands with DMA data in 304
D.1.10 ATAPI Packet commands with DMA data out 305
D.1.11 First-party DMA read of host memory by device 305
D.1.12 First-party DMA write of host memory by device 307
D.1.13 Odd word count considerations 307
D.1.13.1 Legacy DMA read from target for odd word count 308
D.1.13.2 Legacy DMA write by host to target for odd word count 308
D.1.13.3 PIO data read from the device 309
D.1.13.4 PIO data write to the device 309
D.1.13.5 First-party DMA read of host memory by device 310
D.1.13.6 First-party DMA write of host memory by device 310

TABLE OF FIGURES

Figure 1 – Byte, word and Dword relationships 22
Figure 2 – Standard ATA device connectivity 23
Figure 3 – Serial ATA connectivity 24
Figure 4 – Communication layers 25
Figure 5 – Serial ATA connector examples 38
Figure 6 - Signals and grounds assigned in direct connect and cabled configurations 39
Figure 7 – Device plug connector location on 3.5” device 41
Figure 8 – Device plug connector location on 2.5” device 42
Figure 9 – Device plug connector interface dimensions 44
Figure 10 Connector Pin and Feature Locations 46
Figure 11 – Cable receptacle connector interface dimensions 48
Figure 12 – Host plug connector interface dimension 50
Figure 13 – Recommended host plug connector clearance or spacing 51
Figure 14 – Host receptacle connector interface dimensions 53
Figure 15 – Power receptacle connector interface dimensions 55
Figure 16 – Connector pair blind-mate misalignment tolerance 56
Figure 17 - Device-backplane mating configuration 57
Figure 18 – Physical plant overall block diagram 68
Figure 19 – Analog front end (AFE) block diagram 71
Figure 20 – Analog front end (AFE) cabling 73
Figure 21 – Voltage / timing margin base diagram 79
Figure 22 – Jitter output/tolerance mask 80
Figure 23 - Jitter as a function of frequency 82
Figure 24 - Sampling bit error rate formulas 83
Figure 25 - Triangular frequency modulation profile 84
Figure 26 - Spectral fundamental frequency comparison 85
Figure 27 – Common-mode biasing 87
Figure 28 – Out of band signals 89
Figure 29 – Transmitter examples 90
Figure 30 – COMRESET sequence 92
Figure 31 – COMINIT sequence 94
Figure 32 – OOB signal detector 96
Figure 33 – Squelch detector 97
Figure 34 – Low transition density pattern 100
Figure 35 - Half-rate / quarter-rate high transition density pattern 101
Figure 36 – Low frequency spectral content pattern 102
Figure 37 – Simultaneous switching outputs patterns 103
Figure 38 – Composite patterns 104
Figure 39 – Loopback far-end retimed 107
Figure 40 – Loopback far-end analog 108
Figure 41 – Loopback - near-end analog 109
Figure 42 – Compliant test patterns 110
Figure 43 – Jitter measurement example 114
Figure 44 – Signal rise and fall times 115
Figure 45 – Transmit test fixture 115
Figure 46 – Receive test fixture 116
Figure 47 – Power-on Sequence 124
Figure 48 – On to Partial/Slumber - host initiated 127
Figure 49 – ON to Partial/Slumber - device initiated 128
Figure 50 – Bit designations 130
Figure 51 – Nomenclature reference 130
Figure 52 – Conversion examples 131
Figure 53 – Coding examples 134

Figure 54 – Bit ordering and significance 140
Figure 55 – Single bit error with two character delay 142
Figure 56 – Single bit error with one character delay 142
Figure 57 – Transmission structures 144
Figure 58 – CONT usage example 151
Figure 59 – Register - Host to Device FIS layout 186
Figure 60 – Register - Device to Host FIS layout 188
Figure 61 - Set Device Bit - Device to Host FIS layout 190
Figure 62 – DMA Activate - Device to Host FIS layout 191
Figure 63 –DMA Setup – Device to Host FIS layout 192
Figure 64 – BIST Activate - Bidirectional 194
Figure 65 – PIO Setup - Device to Host FIS layout 197
Figure 66 – Data – Host to Device or Device to Host FIS layout 199
Figure 67 – Error handling architecture 272
Figure 68 - Device connector configuration 35A1 292
Figure 69 - Device connector configuration 35B1 292
Figure 70 - Device connector configuration 35B2 293
Figure 71 - Device connector configuration 35B3 293
Figure 72 - Device connector configuration 35B4 293
Figure 73 - Device connector configuration 35B5 294
Figure 74 - Device connector configuration 35C1 294
Figure 75 - Device connector configuration 25A1 294
Figure 76 – Cable construction example 295

TABLE OF TABLES

Table 1 – State Table Cell Description 20
Table 2 – IDENTIFY DEVICE information 33
Table 3 – Device plug connector pin definition 45
Table 4 – Signal integrity requirements and test procedures 59
Table 5 – Housing and contact electrical parameters, test procedures, and requirements 64
Table 6 – Mechanical test procedures, and requirements 65
Table 7 – Environmental parameters, test procedures, and requirements 66
Table 8 – Additional requirement 66
Table 9 – Connector test sequences 67
Table 10. – General system specifications 73
Table 11. – General electrical specifications 76
Table 12. – Voltage / Timing Margin Definition 79
Table 13. – Sampling differential noise budget 81
Table 14 – Desired peak amplitude reduction by SSC. 85
Table 15. – Interface Power States 117
Table 16. – 5B/6B coding 133
Table 17. – 3B/4B coding 133
Table 18. – Valid data characters 135
Table 19. – Valid control characters 139
Table 20. – Description of primitives 146
Table 21 – Primitive encoding 148
Table 22 – SRST write from host to device transmission breaking through a device to host Data FIS 154
Table 23. – Command Shadow Register Block register transmission example 156
Table 24. – Data from host to device transmission example 157
Table 25. – DMA data from host to device, device terminates transmission example 158
Table 26. – Simplified Shadow Register Block register numbering 185
Table 27. – SCR definition 266
Table 28. – SCR Definition 267
Table 29. – CRC and scrambler calculation example - PIO Write Command 289
Table 30. – Type field values 291
Table 31 – Contact material and plating recommendations 296

1 Scope

The purpose of this document is to provide a technical specification of a high-speed serialized ATA data link interface. Normative information is provided to allow a technical design team to construct a device that will operate correctly with another device designed to this specification. Informative information is provided to help illustrate the normative material.

2 Goals, objectives and migration considerations

2.1 Goals and objectives

Serial ATA is defined with the following goals and requirements listed in no particular order:

- Primary inside-the-box storage connection (no outside the box)
- Completely SW transparent w/ ATA (easy transition)
- Low pin count for both host and devices (2 pairs)
- Favorable (low) voltages
- Supports lower cost device architectures
- Higher performance than equivalent ATA (data rate, queuing, overlap) w/ scalability to higher
- Much better cabling/connectors (thin, flexible)
- Includes efficient power delivery
- No software dependency. Relatively easy transition (price, IHV NRE and capital inventory risk, wide variety of devices at intro, etc.)
- Power management and power consumption suitable for mobile use
- Allows roadmap spanning ~10 years
- Cable length comparable to ATA (<1 m)
- Transfer rate exceeding best ATA (~150 MB/s) with scalability to higher rates
- Light protocol allowing overhead latencies to be minimized
- Asynchronous only (no isochronous requirements)
- No Peer-peer transfer support (to/from host only)
- Provides support for 1st party DMA access to host
- Cost competitive with equivalent parallel ATA solution at introduction (host + device + cable)
- Storage device centric (no cameras/scanners/printers)
- Easy installation/configuration (plug/play, no jumpers, no external terminators)
- Single host (no multi-initiators or host/host networking)

2.2 Migration Considerations

The following table illustrates the migration for various aspects of the Serial ATA signaling growth path:

	Generation 1	Generation 2	Generation 3
Approximate speed (8b side)	1.2 Gbits/s (150 Mbytes/sec)	2.4 Gbits/s	4.8[1] Gbits/s
Approximate speed (10b side)	1.5 Gbits/sec	3.0 Gbits/sec	6.0[1] Gbits/sec
Estimated introduction date	mid 2001	mid 2004	mid 2007
Connector		Same as Gen 1	May be upgraded
Cable		Same as Gen 1	May be upgraded
Signaling compatibility		Compatible with Generation 1	Compatible with Generation 2 - may be compatible with Generation 1

NOTE –

1. These speed specifications and schedules are subject to change

3 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.

3.1 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.

AT Attachment with Packet Interface – 5 (ATA/ATAPI-5) [NCITS 340:2000]

To obtain copies of these documents, contact Global Engineering or NCITS.

3.2 Other references

The 8b/10b code used in Serial ATA is based on the following published references

[1] A.X. Widmer and P.A. Franaszek, “A DC-Balanced, Partitioned-Block, 8B/10B Transmission Code.” *IBM Journal of Research and Development*, 27, no. 5: 440-451 (September, 1983)

[2] 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*. (Dec. 4, 1984)

Additional documentation relating to the AT Attachment (ATA) storage interface can be retrieved via the World Wide Web from the Technical Committee T13 at http://www.t13.org.

4 Definitions, abbreviations, and conventions

4.1 Definitions and abbreviations

4.1.1 ATA (AT Attachment)

ATA defines the physical, electrical, transport, and command protocols for the internal attachment of storage devices.

4.1.2 ATAPI (AT Attachment Packet Interface) device

A device implementing the Packet Command feature set.

4.1.3 backchannel

A term used to describe or refer to the transmit same-side SATA interface, when the scope of the paragraph is addressing the receive interface. For example, when discussing the receive serial ATA interface on the device side, the term “backchannel” would be used to describe the transmit interface on the device side.

4.1.4 bit synchronization

The state in which a receiver has synchronized the internal receiver clock to the external transmitter and is delivering retimed serial data.

4.1.5 byte

A byte is 8 bits of data. The least significant bit is bit 0 and the most significant bit is bit 7. The most significant bit is shown on the left. In the encoding process the bits in a byte are referred to as HGFEDCBA, or “A,B,C,D,E,F,G,H” where A corresponds to bit 0 and H corresponds to bit 7. Case is significant.

4.1.6 character

A character is a representation of a data byte or control code. There are two types of characters: data characters and control characters.

4.1.7 character alignment

Character alignment is a receiver action that resets the character boundary to that of the comma sequence found in the K28.5 control character of the ALIGN primitive, and establishes Dword synchronization of the incoming serial data stream.

4.1.8 character slipping

Character slipping is the receiver action that realigns the receiver’s clock to the received bit stream by adding or removing bit times within the characters of the ALIGN primitive.

4.1.9 code violation

A code violation is an error that occurs in the reception process as a result of (1) a running disparity violation or (2) an encoded character that does not translate to a valid data or control character or (3) an encoded character that translates to a control character other than K28.5 or K28.3 in byte 0 of a Dword or (4) an encoded character that translates to any control character (valid or invalid) in bytes 1-3 of a Dword.

4.1.10 comma character

A comma character is a control character, that when encoded, contains the comma sequence. In Serial ATA the only comma character used is K28.5, and only the ALIGN primitive contains the comma character. The comma sequence is the first seven bits of the encoded character.

4.1.11 comma sequence

The comma sequence is a seven-bit sequence of 0011111 or 1100000 in an encoded stream. The comma sequence is unique in that it appears only in a single encoded character, and furthermore, cannot appear in any subset of bits in adjacent encoded characters. This unique property allows the comma sequence to be used for determining alignment of the received data stream.

4.1.12 command aborted

Command aborted is command completion with ABRT set to one in the Error register, and ERR set to one in the Status register.

4.1.13 command completion

Command completion describes the completion of an action requested by command, applicable to the device. Command completion also applies to the case where the command has terminated with an error, and the following actions occurred:

a) the appropriate bits of the Status Register have been updated.
b) BSY & DRQ have been cleared to zero
c) assertion of INTRQ (if nIEN is active-low), assuming that the command protocol specifies INTRQ to be asserted.

In Serial ATA, the register contents are transferred to the host using a Register - Device-to-Host FIS.

4.1.14 command packet

A command packet is a data structure transmitted to the device by a PACKET command that includes the command and command parameters.

4.1.15 Control Block registers

Control Block registers are interface registers used for device control and to post alternate status.

4.1.16 control character

A control character is a combination of a byte value with the control variable equal to K.

4.1.17 control variable

The control variable, Z, is a flag that determines the code set to be used to interpret a data byte. The control variable has the value D (for data characters) or K (for control characters).

4.1.18 CRC

In Serial ATA a 32-bit CRC is calculated over the contents of a Frame Information Structure. The Serial ATA CRC is the Dword in a frame that immediately precedes the EOF primitive.

4.1.19 data character

A data character is a combination of a byte value with the control variable equal to D.

4.1.20 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.

4.1.21 DMA (direct memory access)

Direct memory access is a means of data transfer between device and host memory without host processor intervention.

4.1.22 Dword

A Dword is thirty-two (32) bits of data. A Dword may be represented as 32 bits, as two adjacent words, or as four adjacent bytes. When shown as bits the least significant bit is bit 0 and most significant bit is bit 31. The most significant bit is shown on the left. When shown as words the least significant word (lower) is word 0 and the most significant (upper) word is word 1. When shown as bytes the least significant byte is byte 0 and the most significant byte is byte 3. See Figure 1 for a description of the relationship between bytes, words, and Dwords. Dwords are aligned on four byte boundaries to a zero reference defined by a comma character.

4.1.23 Dword synchronization

The state in which a receiver has recognized the comma sequence and is producing an aligned data stream of Dwords (four contiguous bytes) from the zero-reference of the comma character.

4.1.24 encoded character

An encoded character is the output of the 8b/10b encoder – the result of encoding a character. An encoded character consists of 10 bits, where bit 0 is the most significant bit and bit 9 is the least significant. The bits in an encoded character are symbolically referred to as “abcdeifghj” where “a” corresponds to bit 0 and “j” corresponds to bit 9. Case is significant. Note the out-of-order representation. See Figure 1 for a description of the relationship between bytes, characters and encoded characters.

4.1.25 elasticity buffer

The elasticity buffer is a portion of the receiver where character slipping and/or character alignment is performed.

4.1.26 First-party DMA access

First-party DMA access is a method by which a device accesses host memory.

4.1.27 First-party DMA mode

A device which is operating in First-party DMA mode uses First-party DMA as a primary communications method between the host and the device. A software driver uses legacy mode commands to place the device into First-party DMA mode of operation. The legacy-mode command to place the device into the First-party DMA mode of operation and the command protocol used between a device and host when in First-party DMA mode are not specified by this specification.

4.1.28 FIS

See Frame Information Structure.

4.1.29 Frame Information Structure

The user payload of a frame, does not include the SOF, CRC, and EOF delimiters.

4.1.30 frame

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

4.1.31 interrupt pending

Interrupt pending is an internal state of the device that exists when the device protocol requires the device to notify the host of an event by asserting INTRQ, given the condition where nIEN is asserted active-low to zero.

4.1.32 legacy device

A legacy device is a device which operates in legacy mode.

4.1.33 legacy mode

Legacy mode is the mode of operation which provides software-transparent communication of commands and status between a host and device using the ATA Command Block and Control Block registers.

4.1.34 legal character

A legal character is one for which there exists a valid decoding, either into the data character or control character fields. Due to running disparity constraints not all 10-bit combinations result in a legal character. Additional usage restrictions in Serial ATA result in a further reduction in the SATA defined control character space.

4.1.35 LFSR

See Section 7.4.5

4.1.36 PIO (programmed input/output)

PIO is a means of accessing device registers. PIO is also used to describe one form of data transfers. PIO data transfers are performed by the host processor utilizing PIO register accesses to the Data register.

4.1.37 primitive

A primitive is a single Dword of information that consists of a control character in byte 0 followed by three additional data characters in bytes 1-3.

4.1.38 sector

A sector is a uniquely addressable set of predetermined size, usually 256 words (512 bytes).

4.1.39 Shadow Register Block registers

Shadow Register Block registers are interface registers used for delivering commands to the device or posting status from the device.

4.1.40 unrecoverable error

An unrecoverable error is defined as having occurred at any point when the device sets either the ERR bit or the DF bit to one in the Status register at command completion.

4.1.41 word

A word is sixteen (16) bits of data. A word may be represented as 16 bits or as two adjacent bytes. When shown as bits the least significant bit is bit 0 and most significant bit is bit 15. The most significant bit is shown on the left. When shown as bytes the least significant byte (lower) byte is byte 0 and the most significant byte (upper) byte is byte 1. See Figure 1 for a description of the

relationship between bytes, words and Dwords. The definition of a word in Serial ATA is the same as the definition of a word in ATA.

4.2 Conventions

Lowercase is used for words having the normal English meaning. Certain words and terms used in this document have a specific meaning beyond the normal English meaning. These words and terms are defined either in clause 4.1 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.

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

4.2.1 Precedence

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

4.2.2 Keywords

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

4.2.2.1 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.

4.2.2.2 mandatory

A keyword indicating items to be implemented as defined by this standard.

4.2.2.3 may

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

4.2.2.4 obsolete:

A keyword used to describe bits, bytes, fields, and code values that no longer have consistent meaning or functionality from one implementation to another. However, some degree of functionality may be required for items designated as “obsolete” to provide for backward compatibility. An obsolete bit, byte, field, or command shall never be reclaimed for any other use in any future standard. Bits, bytes, fields, and code values that had been designated as “obsolete” in previous standards may have been reclassified as “retired” in this standard based on the definitions herein for “obsolete” and “retired”.

4.2.2.5 optional

A keyword that describes features that is 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.

4.2.2.6 retired

A keyword indicating that the designated bits, bytes, 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, fields, or code values are utilized before they are reclaimed, they shall have the meaning or functionality as described in previous standards.

4.2.2.7 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 set 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.

4.2.2.8 shall

A keyword indicating a mandatory requirement. Designers are required to implement all such mandatory requirements to ensure interoperability with other standard conformant products.

4.2.2.9 should

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

4.2.3 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.

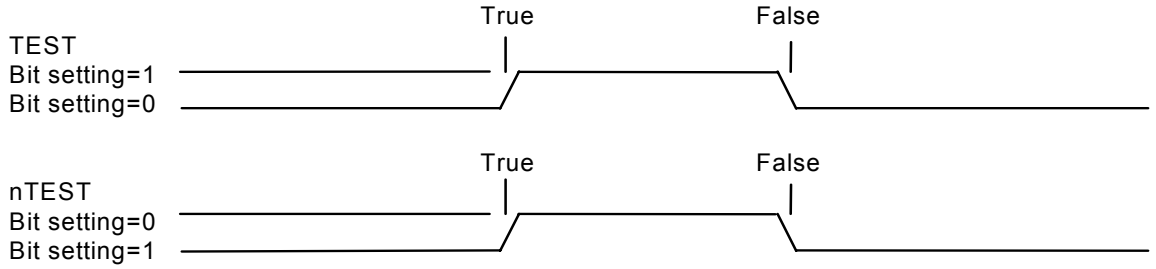
4.2.4 Signal conventions

Signal names are shown in all uppercase letters.

All signals are either high active or low active signals. A Number Sign character (#) at the end of a signal name indicates the signal is a low active signal.

4.2.5 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.



4.2.6 State diagram conventions

For each function to be completed a state machine approach is used to describe the sequence requirements. Each function is composed of several states to accomplish a set goal. Each state of the set is described by an individual state table. Table 1 below shows the general layout for each of the state tables that comprise the set of states for the function.

Table 1 – State Table Cell Description

State name or identifier	Action list[P | W]		
	Branch condition 0	→	**Next state 0**
	Branch condition 1	→	**Next state 1**

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.

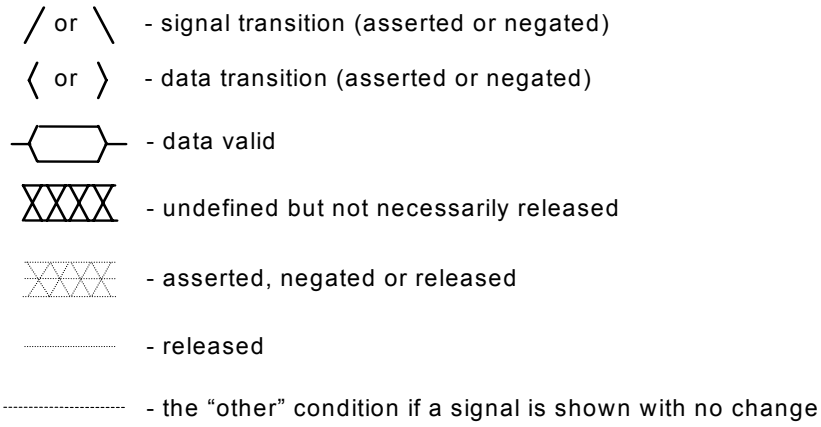
Each transition is identified by a transition label and a transition condition. The transition label consists of the state designator of the state from which the transition is being made followed by the state designator of the state to which the transition is being made. In some cases, the transition 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 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.

It is assumed that all actions are executed within a state and that transitions from state to state are instantaneous.

4.2.7 Timing conventions

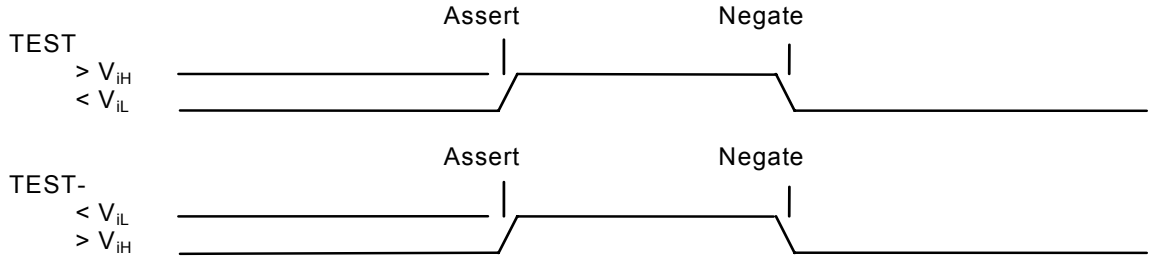
Certain symbols are used in the timing diagrams. These symbols and their respective definitions are listed below.



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.



In the case of the Serial ATA interface itself, all signals are differential. Each signal is actually transmitted or received on two lines and the state of the signal is determined by the relationship of these lines to each other rather than to a reference ground. These two line are named the same except the names end in either “+” or “-“. The above symbols are used when representing these signals however, assertion of a differential signal is assumed to mean that the “+” signal line is positive in relation to the “-“ line. In other words, if the signal is “TX”, assertion means that the voltage on TX+ minus the voltage on TX- results in a positive number. Conversely, negation means that the result is a negative number.

4.2.8 Byte, word and Dword Relationships

Figure 1 – Byte, word and Dword relationships illustrates the relationship between bytes, words and Dwords.

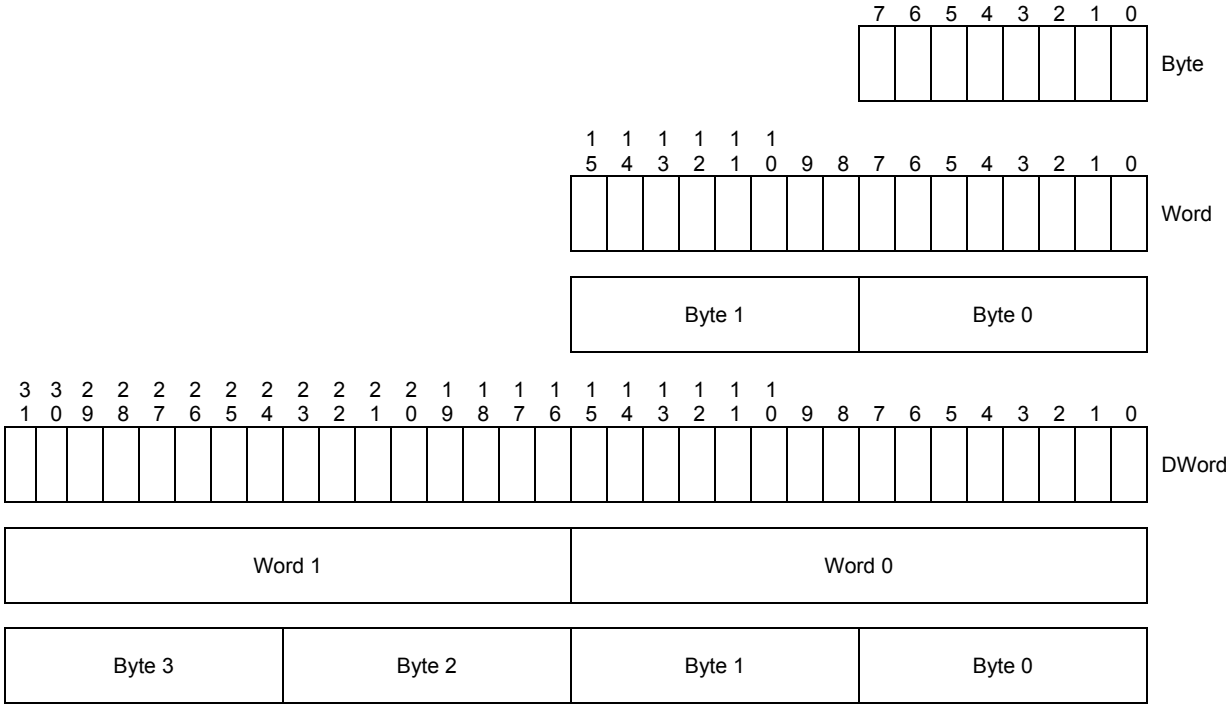


Figure 1 – Byte, word and Dword relationships

5 General overview

Serial ATA is a high-speed serial link replacement for the parallel ATA attachment of mass storage devices. The serial link employed is a high-speed differential layer that utilizes Gigabit technology and 8b/10b encoding.

Figure 2 – Standard ATA device connectivity illustrates how two devices are connected to a standard ATA host adapter. This method allows up to two devices to be connected to a single port using a Master/Slave communication technique. Each device is connected via a ribbon cable that “daisy chains” the devices.

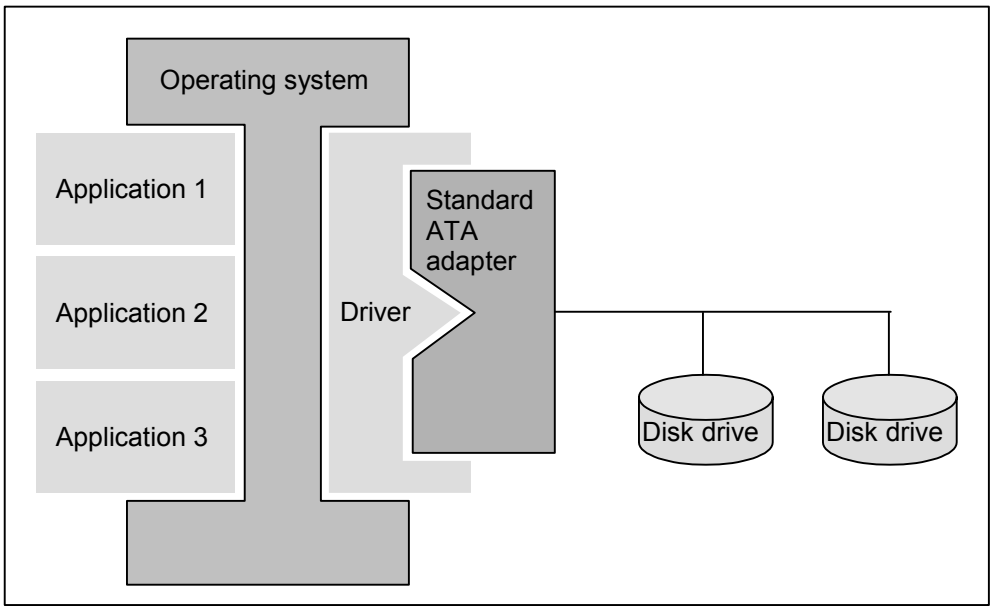


Figure 2 – Standard ATA device connectivity

Figure 3 – Serial ATA connectivity illustrates how the same two devices are connected using a Serial ATA host adapter. In this diagram the dark grey portion is functionally identical to the dark grey portion of the previous diagram. Legacy host software that is only ATA aware accesses the Serial ATA subsystem in the exact same manner and will function correctly. In this case, however, the software views the two devices as if they were “masters” on two separate ports. The right hand portion of the host adapter is of a new design that converts the normal operations of the software into a serial data/control stream. The Serial ATA structure connects each of the two drives with their own respective cables in a point-to-point fashion.

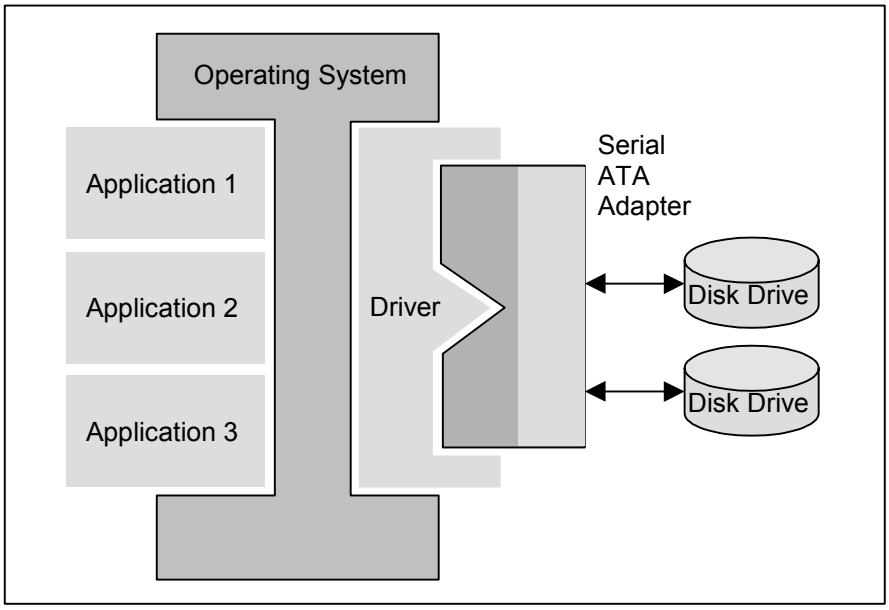


Figure 3 – Serial ATA connectivity

5.1 Sub-module operation

The Transport control state machine and the Link state machine are the two core sub-modules that control overall operation. The Link state machine controls the operation(s) related to the serial line and the Transport control state machine controls the operation(s) relating to the host platform. The two state machines coordinate their actions and utilize resources to transfer data between a host computer and attached mass storage device. The host Link state machine communicates via the serial line to a corresponding Link state machine located in the device. The host Transport machine also likewise communicates with a corresponding device Transport state machine. The two Link state machines ensure that control sequences between the two Transport control state machines are properly exchanged. Figure 4 shows how the machines communicate their various needed parameters in the traditional layered model. Each layer communicates with its counterpart directly or indirectly

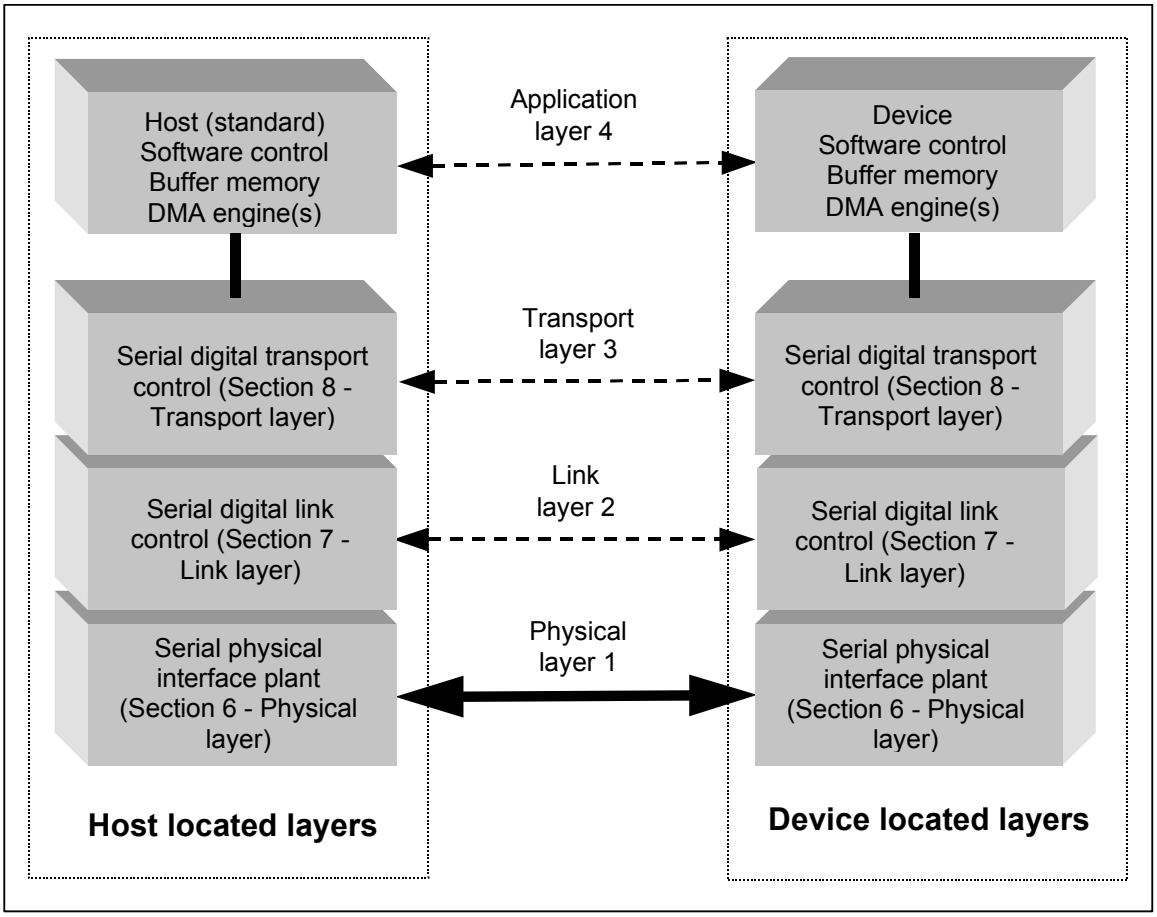


Figure 4 – Communication layers

The host interacts with the Transport control state machine through a register interface that is equivalent to that presented by a traditional parallel ATA host adapter. The host software follows existing standards and conventions when accessing the register interface and follows standard command protocol conventions. The Transport control state machine breaks down these operations into a sequence of actions that are exchanged with the Link state machine. The Transport control state machine uses resources located in the ATA interface sub-module to perform its operations.

5.2 Standard ATA Emulation

Emulation of parallel ATA device behavior as perceived by the host BIOS or software driver, is a cooperative effort between the device and the Serial ATA host adapter hardware. The behavior of Command and Control Block registers, PIO and DMA data transfers, resets, and interrupts are all emulated.

The host adapter contains a set of registers that shadow the contents of the traditional device registers, referred to as the Shadow Register Block. All Serial ATA devices behave like Device 0 devices. Devices shall ignore the DEV bit in the Device/Head field of received Register FIS's, and it is the responsibility of the host adapter to gate transmission of Register FIS's to devices, as appropriate, based on the value of the DEV bit.

After a reset or power-on, the host bus adapter emulates the behavior of a traditional ATA system during device discovery. Immediately after reset, the host adapter shall place the value 0x7Fh in its Status shadow register and shall place the value 0xFFh in all the other taskfile registers (0xFFFFh in the Data register). In this state the host bus adapter shall not accept writes to the taskfile shadow registers. When the Phy detects presence of an attached device, the host bus adapter shall now set bit 7 in the Status shadow register yielding the value 0xFFh, and the host bus adapter shall now allow writes to the shadow taskfile registers. If a device is present, the Phy shall take no longer than 10 ms to indicate that it has detected the presence of a device, and has set bit 7 in the Status shadow register. When the attached device establishes communication with the host bus adapter, it shall send a register FIS to the host, resulting in the shadow registers being updated with values appropriate for the attached device.

The host adapter may present a Master-only emulation to host software, that is, each device is a Device 0, and each Device 0 is accessed at a different set of host bus addresses. The host adapter may optionally present a Master/Slave emulation to host software, that is, two devices on two separate Serial ATA ports are represented to host software as a Device 0 and a Device 1 accessed at the same set of host bus addresses.

5.2.1 Software Reset

According to the parallel ATA specification, issuing a software reset is done by toggling the SRST bit in the Control register. The toggle period is stipulated as being no shorter than 5us. As a result of the SRST bit changing in the Control register, Serial ATA host adapters shall issue at least two Register FISes to the device (one with the SRST bit asserted and a subsequent one with the SRST bit cleared). See section 8.5.2 for a detailed definition of Register FIS. Although host software is required to toggle the SRST bit no faster than 5us, devices may not rely on the inter-arrival time of received Register FISes also meeting this timing. Because of flow control, frame handshaking, and other protocol interlocks, devices may effectively receive the resulting Register FISes back-to-back.

Due to flow control, protocol interlocks, power management state, or other transmission latencies, the subsequent Register frame transmission clearing the SRST bit during a software reset may be triggered prior to the previous Register FIS transmission having been completed. Host adapters are required to allow host software to toggle the SRST with the minimum 5us timing specified in the parallel ATA specification even if frame transmission latencies result in the first Register FIS transmission taking longer than 5us. Host adapters are required to ensure transmission of the two resulting Register FISes to the device regardless of the transmission latency of each individual FIS.

5.2.2 Master-only emulation

A native Serial ATA host adapter behaves the same as if a legacy master only device were attached with no slave present. It is the responsibility of the host adapter to properly interact with host software and present the correct behavior for this type of configuration. All Serial ATA devices, therefore, need not be aware of master / slave issues and ignore legacy task file information that deal with a

secondary device. When the DEV bit in the Device/Head register is set to one, selecting the non-existent Device 1, the host adapter shall respond to register reads and writes as specified for a Device 0 with no Device 1 present, as defined in the ATA standard. This includes not setting the BSY bit in the shadow Status register when Device 1 is selected, as described in the ATA standard. When Device 0 is selected, the host adapter shall execute the Serial ATA protocols for managing the shadow register contents as defined in later sections.

When Device 0 is selected and the Command register is written in the Shadow Register Block, the host adapter sets the BSY bit in its shadow Status register. The host adapter then transmits a frame to the device containing the new register contents. When the Device Control register is written in the Shadow Register Block with a change of state of the SRST bit, the host adapter sets the BSY bit in its shadow Status register and transmit a frame to the device containing the new register contents. Transmission of register contents when the Device Control register is written with any value that is not a change of state of the SRST bit shall not set the BSY bit in the shadow Status register, and transmission of a frame to the device containing new register contents is optional. Similarly, the host adapter sets the BSY bit in its shadow Status register to one when a COMRESET is requested or the SRST bit is set to one in the Device Control register. The expected timing for BSY bit assertion is thereby preserved.

The device updates the contents of the host adapter Shadow Register Block by transmitting a register frame to the host. This allows the device to set the proper ending status in the host adapter Shadow Register Block at the completion of a command or control request. Specific support is added to ensure proper timing of the DRQ and BSY bits in the Status register for PIO transfers.

Finally the host adapter cooperates with the device by providing an interrupt pending flag in the host adapter. This flag is set by the host adapter when the device sends a serial bus frame including the request to set the interrupt pending flag. The host adapter asserts the interrupt to the host processor any time the interrupt pending flag is set, the DEV bit is cleared to zero in the shadow Device/Head register, and the nIEN bit in the shadow Device Control register is cleared to zero. The host adapter clears the interrupt pending flag any time a COMRESET is requested, the SRST bit in the shadow Device Control register is set to one, the shadow Command register is written with DEV cleared to zero, or the shadow Status register is read with DEV cleared to zero. This allows the emulation of the host interrupt and its proper timing.

5.2.3 Master/Slave emulation (optional)

All devices behave as if they are Device 0 devices. However, the host adapter may optionally implement emulation of the behavior of a Master/Slave configuration by pairing two Serial ATA ports, and managing their associated shadow registers accordingly, as though they were a Device 0 and Device 1 at the same set of host bus addresses.

A host adapter that emulates Master/Slave behavior shall manage the two sets of shadow registers (one set for each of the two devices) based on the value of the DEV bit in the shadow Device/Head register. Based on the value of the DEV bit, the host adapter shall direct accesses to the Shadow Register Block Registers, and accesses to the Control Block Registers, to the appropriate set of shadow registers in the correct device. It is the responsibility of the host adapter to ensure that communication with one or both of the attached devices is handled properly, and that information gets routed to the devices correctly. Each device shall process any communication with the host adapter as if it is targeted for the device regardless of the value of the DEV bit.

If a host adapter is emulating Master/Slave behavior, and there is no device attached to the cable designated as the Device 1 cable, the host adapter shall emulate Device 0 behavior with no Device 1 present as described in the ATA standard.

5.2.3.1 Software reset

Host adapters that emulate Master/Slave behavior shall emulate proper behavior for software reset. Based on the Phy initialization status, the host adapter knows whether a device is attached to each of the two ports used in a Master/Slave emulation configuration. Device Control Register writes, that have the SRST bit set to one, shall result in the associated shadow register being written for each port to which a device is attached. The frame transmission protocol for each associated port executed, results in a Register FIS being transmitted to each attached device. Similarly, the subsequent write to the Device Control register that clears the SRST bit shall result in a Register FIS being sent to each attached device. The host adapter shall then await a response from each attached device (or timeout), and shall merge the contents of the Error and Status registers for the attached devices, in accordance with the ATA standard, to produce the Error and Status register values visible to host software.

5.2.3.2 EXECUTE DEVICE DIAGNOSTICS

Host adapters that emulate Master/Slave behavior shall emulate proper behavior for EXECUTE DEVICE DIAGNOSTICS. The host adapter shall detect the EXECUTE DEVICE DIAGNOSTIC command being written to the Command register. Detecting the EXECUTE DEVICE DIAGNOSTICS command shall result in the associated shadow register being written for each port to which a device is attached. The frame transmission protocol for each associated port executed, results in a Register FIS being transmitted to each attached device. The host adapter shall then await response from each attached device (or timeout), and shall merge the contents of the Error and Status registers for the attached devices, in accordance with the ATA standard to produce the Error and Status register values visible to host software.

5.2.3.3 Restrictions and limitations

Superset capabilities that are unique to Serial ATA and not supported by legacy parallel ATA are not required to be supported in Master/Slave emulation mode. Such capabilities include support for first-party DMA, dynamic plug/unplug, fine-grained interface power management, and superset status and control information. Master/Slave emulation is recommended only in configurations where legacy software drivers are used and the number of attached devices exceeds the number of taskfile interfaces the legacy software supports.

5.2.4 Standard ATA interoperability state diagrams

This state diagram defines the protocol of the host adapter to emulate Master only legacy ATA devices as seen from the host BIOS or software driver. The interrupt pending flag (IPF) is an internal state bit in the host adapter that reflects whether or not the device has an interrupt pending to the host.

HA0: HA_SEL/NOINTRQ	The interrupt signal is not asserted to the host.		
1.	COMRESET requested by the host.	→	HA_SEL/NOINTRQ
2.	Device Control register written by the host with SRST=0 and (nIEN=1 or IPF=0).	→	HA_SEL/NOINTRQ
3.	Device Control register written by the host with SRST=0 and (nIEN=0 and IPF=1).	→	HA_SEL/INTRQ
4.	Device Control register written by the host with SRST=1.	→	HA_SEL/NOINTRQ
5.	Command register written by the host.	→	HA_SEL/NOINTRQ
6.	Device/Head register written by the host with DEV = 1.	→	HA_NOTSEL
7.	Any other register read or write by the host	→	HA_SEL/NOINTRQ
8.	Transport layer indicates new register content from the device with no set IPF flag.	→	HA_SEL/NOINTRQ
9.	Transport layer indicates new register content from the device with set IPF flag and nIEN=1.	→	HA_SEL/NOINTRQ
10.	Transport layer indicates new register content from the device with set IPF flag and nIEN=0.	→	HA_SEL/INTRQ

HA1: HA_SEL/INTRQ	The interrupt signal is asserted to the host.		
1.	COMRESET requested by the host.	→	HA_SEL/NOINTRQ
2.	Device Control register written by the host with SRST=0 and (nIEN=1 or IPF=0).	→	HA_SEL/NOINTRQ
3.	Device Control register written by the host with SRST=0 and (nIEN=0 and IPF=1).	→	HA_SEL/INTRQ
4.	Device Control register written by the host with SRST=1.	→	HA_SEL/NOINTRQ
5.	Command register written by the host.	→	HA_SEL/NOINTRQ
6.	Device/Head register written by the host with DEV = 1.	→	HA_NOTSEL
7.	Status register read by the host	→	HA_SEL/NOINTRQ
8.	Any other register write by the host	→	HA_SEL/INTRQ
9.	Any other register read by the host	→	HA_SEL/INTRQ
10.	Transport layer indicates new register content from the device.	→	HA_SEL/INTRQ

HA2: HA_NOTSEL	The interrupt signal is not asserted to the host.	
1. COMRESET requested by the host.	→	HA_SEL/NOINTRQ
2. Device Control register written by the host with SRST = 0.	→	HA_NOTSEL
3. Device Control register written by the host with SRST = 1.	→	HA_SEL/NOINTRQ
4. Command register written by the host with command other than EXECUTE DEVICE DIAGNOSTIC.	→	HA_NOTSEL
5. Command register written by the host with command EXECUTE DEVICE DIAGNOSTIC.	→	HA_SEL/NOINTRQ
6. Device/Head register written by the host with DEV = 0 and (nIEN=1 or IPF=0).	→	HA_SEL/NOINTRQ
7. Device/Head register written by the host with DEV = 0 and (nIEN=0 and IPF=1).	→	HA_SEL/INTRQ
8. Any other register write by the host	→	HA_NOTSEL
9. Read of Status or Alternate Status register by the host.	→	HA_NOTSEL
10. Read of any other register by the host.	→	HA_NOTSEL
11. Transport layer indicates new register content from the device.	→	HA_NOTSEL

HA0: HA_SEL/NOINTRQ state: This state is entered when Device 0 is selected and either IPF is cleared to zero or nIEN is set to one.

When in this state, the interrupt signal to the host shall not be asserted.

Transition HA0:1: When a COMRESET is requested by the host, the host adapter shall set BSY to one in the shadow Status register, clear IPF to zero, notify the Link layer to have a bus COMRESET asserted, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA0:2: When the Device Control register is written by the host with SRST cleared to zero and either nIEN set to one or IPF is cleared to zero, the host adapter shall notify the Transport layer to send a ContrlFIS with the current content of the shadow registers, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA0:3: When the Device Control register is written by the host with SRST cleared to zero, nIEN cleared to zero, and IPF is set to one, the host adapter shall notify the Transport layer to send a ContrlFIS with the current content of the shadow registers, and make a transition to the HA1: HA_SEL/INTRQ state.

Transition HA0:4: When the Device Control register is written by the host with SRST set to one, the host adapter shall set BSY to one in the shadow Status register, clear IPF to zero, notify the Transport layer to send a ContrlFIS with the current content of the shadow registers, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA0:5: When the Command register is written by the host, the host adapter shall set BSY to one in the shadow Status register, clear IPF to zero, notify the Transport layer to send a CmdFIS with the current content of the shadow registers, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA0:6: When the Device/Head register is written by the host with DEV set to one, the host adapter shall make a transition to the HA2: HA_NOTSEL state.

Transition HA0:7: When any register is written by the host other than those described above, the host adapter shall make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA0:8: When the Transport layer indicates new register content from the device with no set IPF flag, the host adapter shall place the new register content into the shadow registers and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA0:9: When the Transport layer indicates new register content from the device with set IPF flag and nIEN is set to one, the host adapter shall set IPF to one, place the new register content into the shadow registers, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA0:10: When the Transport layer indicates new register content from the device with set IPF flag and nIEN is cleared to zero, the host adapter shall set IPF to one, place the new register content into the shadow registers, and make a transition to the HA1: HA_SEL/INTRQ state.

HA1: HA_SEL/INTRQ state: This state is when DEVICE 0 is selected, nIEN is cleared to zero, and IPF is set to one.

When in this state, the interrupt signal to the host shall be asserted.

Transition HA1:1: When a COMRESET is requested by the host, the host adapter shall set BSY to one in the shadow Status register, clear IPF to zero, notify the Link layer to have a bus COMRESET asserted, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA1:2: When the Device Control register is written by the host with SRST cleared to zero, with nIEN set to one or IPF cleared to zero, the host adapter shall notify the Transport layer to send a ContrlFIS with the current content of the shadow registers, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA1:3: When the Device Control register is written by the host with SRST cleared to zero with nIEN cleared to zero and IPF set to one, the host adapter shall notify the Transport layer to send a ContrlFIS with the current content of the shadow registers, and make a transition to the HA1: HA_SEL/INTRQ state.

Transition HA1:4: When the Device Control register is written by the host with SRST set to one, the host adapter shall set BSY to one in the shadow Status register, clear IPF to zero, notify the Transport layer to send a ContrlFIS with the current content of the shadow registers, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA1:5: When the Command register is written by the host, the host adapter shall set BSY to one in the shadow Status register, clear IPF to zero, notify the Transport layer to send a CmdFIS with the current content of the shadow registers, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA1:6: When the Device/Head register is written by the host with DEV set to one, the host adapter shall make a transition to the HA2: HA_NOTSEL state.

Transition HA1:7: When the Status register is read by the host, the host adapter shall clear IPF to zero and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA1:8: When any register is written by the host other than those described above, the host adapter shall make a transition to the HA1: HA_SEL/INTRQ state.

Transition HA1:9: When any register is read by the host other than that described above, the host adapter shall make a transition to the HA1: HA_SEL/INTRQ state.

Transition HA1:10: When the Transport layer indicates new register content from the device, the host adapter shall place the new register content into the shadow registers and make a transition to the HA1: HA_SEL/INTRQ state.

HA2: HA_NOTSEL state: This state is entered when DEVICE 1 is selected.

When in this state, the interrupt signal to the host shall not be asserted.

Transition HA2:1: When a COMRESET is requested by the host, the host adapter shall set BSY to one in the shadow Status register, clear DEV to zero in the shadow Device/Head register, clear IPF to zero, notify the Link layer to have a bus COMRESET asserted, and make a transition to the HA0: SEL/NOINTRQ state.

Transition HA2:2: When the Device Control register is written by the host with SRST cleared to zero, the host adapter shall notify the Transport layer to send a ContrlFIS with the current content of the shadow registers, and make a transition to the HA2: HA_NOTSEL state.

Transition HA2:3: When the Device Control register is written by the host with SRST set to one, the host adapter shall set BSY to one in the shadow Status register, clear DEV to zero in the shadow Device/Head register, clear IPF to zero, notify the Transport layer to send a ContrlFIS with the current content of the shadow registers, and make a transition to the HA0: HA_ SEL/NOINTRQ state.

Transition HA2:4: When the Command register is written by the host with any command code other than EXECUTE DEVICE DIAGNOSTIC, the host adapter shall not set BSY in the shadow Status register and shall make a transition to the HA2: HA_NOTSEL state.

Transition HA2:5: When the Command register is written by the host with the EXECUTE DEVICE DIAGNOSTIC command code, the host adapter shall set BSY to one in the shadow Status register, clear DEV to zero in the Device/Head register, clear IPF to zero, notify the Transport layer to send a CmdFIS with the current content of the shadow registers, and make a transition to the HA0: HA_SEL/INTRQ state.

Transition HA2:6: When the Device/Head register is written by the host with DEV cleared to zero and either nIEN is set to one or IPF is cleared to zero, the host adapter shall make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA2:7: When the Device/Head register is written by the host with DEV cleared to zero, nIEN cleared to zero, and IPF is set to one, the host adapter shall make a transition to the HA1: HA_SEL/INTRQ state.

Transition HA2:8: When any register is written by the host other than those described above, the host adapter shall make a transition to the HA2: HA_NOTSEL state.

Transition HA2:9: When the Status or Alternate Status register is read by the host, the host shall return register content 00h to the host and make a transition to the HA2: HA_NOTSEL state.

Transition HA2:10: When any register is read by the host other than that described above, the host adapter shall return the current shadow register content to the host and make a transition to the HA2: HA_NOTSEL state.

Transition HA2:11: When the Transport layer indicates new register content from the device, the host adapter shall place the new register content into the shadow registers and make a transition to the HA2: HA_NOTSEL state.

5.2.5 IDENTIFY DEVICE command

In the IDENTIFY DEVICE command various parameters are communicated to the host from the device. The following sections define those words that are different from and additions to the ATA/ATAPI-5 standard definition of the data contents.

Table 2 – IDENTIFY DEVICE information

Word	F/V		
0-46		Set as indicated in ATA/ATAPI-5	
47			
	F	15-8	80h
	R	7-0	00h = Reserved 01h-FFh = Maximum number of sectors that shall be transferred per interrupt on READ/WRITE MULTIPLE commands
48		Set as indicated in ATA/ATAPI-5	
49		Capabilities	
	R	15-14	Reserved for the IDENTIFY PACKET DEVICE command.
	F	13	1=Standby timer values as specified in this standard are supported 0=Standby timer values shall be managed by the device
	R	12	Reserved for the IDENTIFY PACKET DEVICE command.
	F	11	1=IORDY supported 0=IORDY may be supported
	F	10	1=IORDY may be disabled
	R	9	Shall be set to one. Utilized by IDENTIFY PACKET DEVICE command.
	R	8	Shall be set to one. Utilized by IDENTIFY PACKET DEVICE command.
	X	7-0	Retired
50-52		Set as indicated in ATA/ATAPI-5	
53	R	15-3	Reserved
	F	2	1=the fields reported in word 88 are valid 0=the fields reported in word 88 are not valid
	F	1	1=the fields reported in words 64-70 are valid 0=the fields reported in words 64-70 are not valid
	V	0	1=the fields reported in words 54-58 are valid 0=the fields reported in words 54-58 may be valid
54-62		Set as indicated in ATA/ATAPI-5	
63		15-3	Set as indicated in ATA/ATAPI-5
	F	2	1= Multiword DMA mode 2 and below are supported
	F	1	1= Multiword DMA mode 1 and below are supported
	F	0	1= Multiword DMA mode 0 is supported
64		15-2	Set as indicated in ATA/ATAPI-5
	F	1-0	PIO modes 3 and 4 supported
65		Minimum Multiword DMA transfer cycle time per word	
	F	15-0	Cycle time in nanoseconds
66		Manufacturer's recommended Multiword DMA transfer cycle time	
	F	15-0	Cycle time in nanoseconds
67		Minimum PIO transfer cycle time without flow control	
	F	15-0	Cycle time in nanoseconds
68		Minimum PIO transfer cycle time with IORDY flow control	
	F	15-0	Cycle time in nanoseconds
69-87		Set as indicated in ATA/ATAPI-5	
88		15-5	Set as indicated in ATA/ATAPI-5
	F	4	1=Ultra DMA mode 4 and below are supported
	F	3	1=Ultra DMA mode 3 and below are supported
	F	2	1=Ultra DMA mode 2 and below are supported

	F F	1 1=Ultra DMA mode 1 and below are supported 0 1=Ultra DMA mode 0 is supported
89-92		Set as indicated in ATA/ATAPI-5
93	V	COMRESET result. The contents of this word shall be set to zero.
94-255		Set as indicated in ATA/ATAPI-5

Key:
F = the content of the word is fixed and does not change. For removable media devices, these values may change when media is removed or changed.
V = the contents of the word is variable and may change depending on the state of the device or the commands executed by the device.
X = the content of the word is vendor specific and may be fixed or variable.
R = the content of the word is reserved and shall be zero.

5.2.5.1 Word 0 - 46: Set as indicated in ATA/ATAPI-5

5.2.5.2 Word 47: Multiword PIO transfer

Bits 15 through 8 of word 47 shall be set as indicated in ATA/ATAPI-5.
Bits 7 through 0 are used to indicate the maximum number of sectors that shall be transferred per interrupt on READ/WRITE MULTIPLE commands. This field should be set to 16 or less. See section 8.5.8.1.

5.2.5.3 Word 48: Set as indicated in ATA/ATAPI-5

5.2.5.4 Word 49: Capabilities

Bits 15 through 12 of word 49 shall be set as indicated in ATA/ATAPI-5.

Bit 11 of word 49 is used to determine whether a device supports IORDY. This bit shall be set to one, indicating the device supports IORDY operation.

Bit 10 of word 49 is used to indicate a device's ability to enable or disable the use of IORDY. This bit shall be set to one, indicating the device supports the disabling of IORDY. Disabling and enabling of IORDY is accomplished using the SET FEATURES command.

Bits 9 through 0 of word 49 shall be set as indicated in ATA/ATAPI-5.

5.2.5.5 Words 50 - 52: Set as indicated in ATA/ATAPI-5

5.2.5.6 Word 53: Field validity

Bit 0 of word 53 shall be set to one, indicating the values reported in words 54 through 58 are valid. Bit 1 of word 53 shall be set to one, the values reported in words 64 through 70 are valid. Any device that supports PIO mode 3 or above, or supports Multiword DMA mode 1 or above, shall set bit 1 of word 53 to one and support the fields contained in words 64 through 70. Bit 2 of word 53 shall be set to one indicating the device supports Ultra DMA and the values reported in word 88 are valid. Bits 15-3 are reserved.

5.2.5.7 Word 54 - 62: Set as indicated in ATA/ATAPI-5

5.2.5.8 Word 63: Multiword DMA transfer

Bits 2 - 0 of word 63 shall be set to one indicating that the device supports Multiword DMA modes 0, 1, and 2. Bits 15 – 3 shall be set as indicated in ATA/ATAPI-5.

5.2.5.9 Word 64: PIO transfer modes supported

Bits 1 - 0 of word 64 shall be set to one indicating that the device supports PIO modes 3 and 4. Bits 15 – 2 shall be set as indicated in ATA/ATAPI-5.

5.2.5.10 Word 65: Minimum Multiword DMA transfer cycle time per word

Shall be set to indicate 120 ns.

5.2.5.11 Word 66: Device recommended Multiword DMA cycle time

Shall be set to indicate 120 ns.

5.2.5.12 Word 67: Minimum PIO transfer cycle time without flow control

Shall be set to indicate 120 ns.

5.2.5.13 Word 68: Minimum PIO transfer cycle time with IORDY

Shall be set to indicate 120 ns.

5.2.5.14 Words 69-87: Set as indicated in ATA/ATAPI-5

5.2.5.15 Word 88: Ultra DMA modes

Bits 4 - 0 of word 88 shall be set to one indicating that the device supports Ultra DMA modes 0, 1, 2, 3, and 4. Bits 15 – 5 shall be set as indicated in ATA/ATAPI-5.

5.2.5.16 Word 89 - 92: Set as indicated in ATA/ATAPI-5

5.2.5.17 Word 93: Hardware configuration test results

Word 93 shall be set to 0000h indicating that the word is not supported.

5.2.5.18 Words 94-255: Set as indicated in ATA/ATAPI-5

6 Physical layer

6.1 Overview

This section describes the physical layer of Serial ATA. The information that is provided is comprised of two types – informative and normative. Unless otherwise described, the information should be considered normative in nature and is included in this document as a necessary requirement in order to properly allow a piece of equipment to attach to another piece of equipment. The normative information is deliberately structured to constrain and define areas only to the degree that is required for compatibility. The information that is provided and marked informative is provided only to help the reader better understand the normative sections and should be taken as examples only. Exact implementations may vary.

6.2 List of services

- Transmit a 1.5 Gb/sec differential NRZ serial stream at specified voltage levels
- Provide a 100 Ohm matched termination (differential) at the transmitter
- Serialize a 10, 20, 40, or other width parallel input from the Link for transmission
- Receive a 1.5 Gb/sec +350/- 2650 ppm differential NRZ serial stream
- Provide a 100 Ohm matched termination (differential) at the receiver
- Extract data (and, optionally, clock) from the serial stream
- Deserialize the serial stream
- Detect the K28.5 comma character and provide a bit and word aligned 10, 20, 40, or other width parallel output
- Provide specified OOB signal detection and transmission
- Perform proper power-on sequencing and speed negotiation
- Provide device status to Link layer
  - device present
  - device absent
  - device present but failed to negotiate communications
- Optionally support power management modes (note: if not supported, power management requests must be properly rejected by the Link layer)
- Optionally perform transmitter and receiver impedance calibration (if manufacturing tolerances require)
- Handle the input data rate frequency variation due to a spread spectrum transmitter clock (Nominal +0/-0.5% slow frequency variation, over a 33.33 us up/down triangular wave period).
- Accommodate request to go into Far-End retimed loopback test mode of operation when commanded (exit this mode on OOB COMRESET/COMINIT)

6.3 Cables and connectors specifications

6.3.1 Overview

This section covers Serial ATA connectors and cable assemblies. It defines the

- Connector mating interfaces
- Connector location on the device
- Electrical, mechanical and reliability requirements of the connectors and cable assemblies
- Connector and cable testing procedures

It, however, does not define how the connector and cable assembly should be implemented, such as

- The mounting feature of the connectors
- The cabling and cable terminations
- The methods on how the PCB connects to other components of the system

6.3.2 Objectives

The basic guidelines for Serial ATA connectors and cables are

- Supporting 1.5 Gbps date rate with headroom for 3.0 Gbps speed
- Cost competitive to Ultra ATA connector and cable
- Facilitating smooth transition from Ultra ATA to Serial ATA
- Common connector interface for both 2.5” and 3.5” devices

From those guidelines, the following connector and cable objectives are derived:

- Minimal discontinuity at connectors
- Good impedance control for cable
- 0 to 1 meter cable length
- Low profile, fitting in the 2.5” drive
- Blind-mateable
- Hot-pluggable by means of staggered contacts
- Supporting power delivery with 12.0 V, 5.0 V, and 3.3 V voltages

6.3.3 General descriptions

A Serial ATA device may be either directly connected to a host or connected to a host through a cable.

For direct connection, the device plug connector, shown as (a) and (b) in Figure 5, is inserted directly into a host receptacle connector, illustrated as (g) in Figure 5. The device plug connector and the host receptacle connector incorporate features that enable the direct connection to be hot pluggable and blind mateable.

For connection via cable, the device signal plug connector, shown as (a) in Figure 5, mates with the signal cable receptacle connector on one end of the cable, illustrated as (c) in Figure 5. The signal cable receptacle connector on the other end of the cable is inserted into a host signal plug connector, shown as (f) in Figure 5. The signal cable wire consists of two twinax sections in a common outer sheath.

Besides the signal cable, there will be also a separate power cable for the cabled connection. A Serial ATA power cable includes a power cable receptacle connector, shown as (d) in Figure 5, on one end and may be directly connected to the host power supply on the other end or may include a power cable receptacle on the other end. The power cable receptacle connector on one end of the

power cable mates with the device power plug connector, shown as (b) in Figure 5. The other end of the power cable is attached to the host as and however necessary.

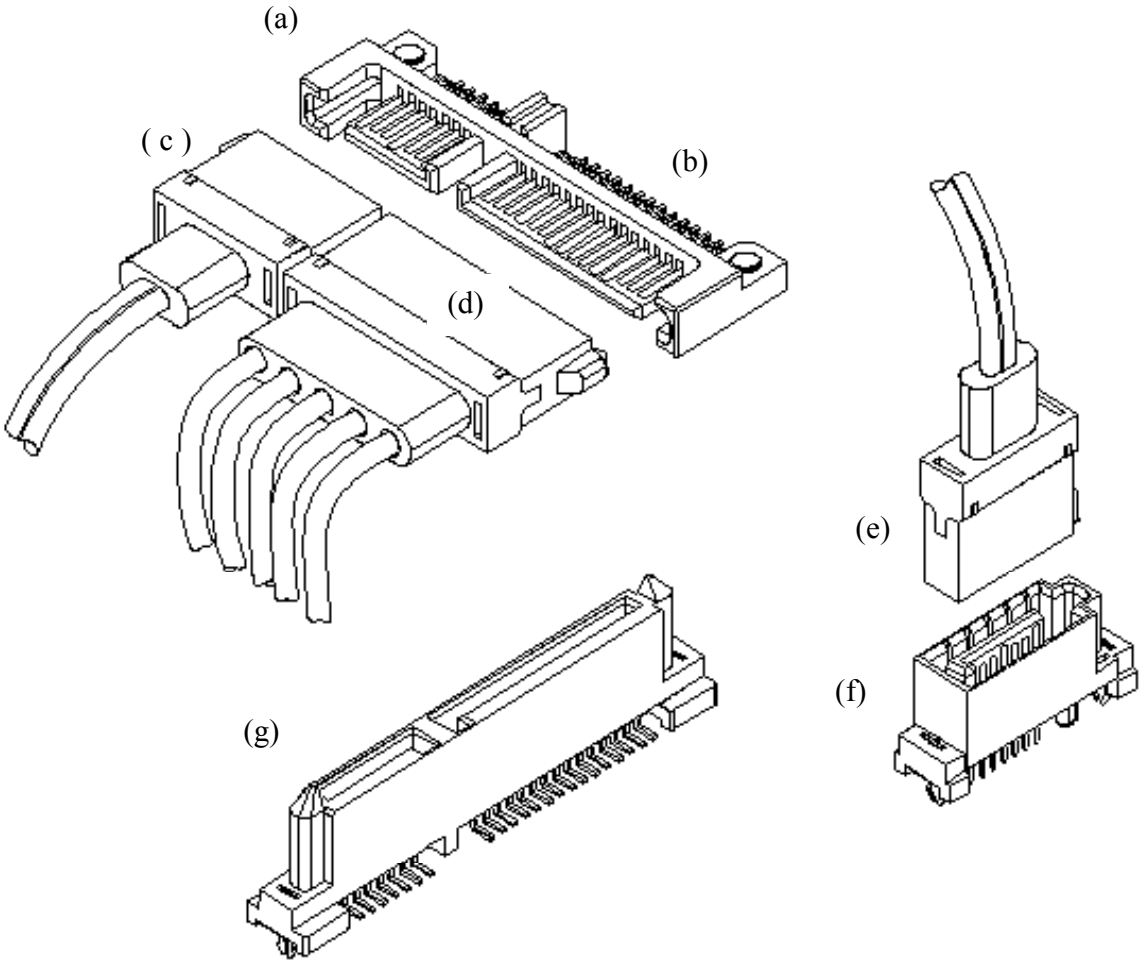


Figure 5 – Serial ATA connector examples

Illustrated above: (a) device signal plug segment or connector; (b) device power plug segment or connector; (c) signal cable receptacle connector, to be mated with (a); (d) power cable receptacle connector, to be mated with (b); (e) signal cable receptacle connector, to be mated with (f), the host signal plug connector; (g) host receptacle connector mating directly with device plug connector (a) & (b).

Figure 6 illustrates how signals and grounds are assigned in direct connect and cabled configurations.

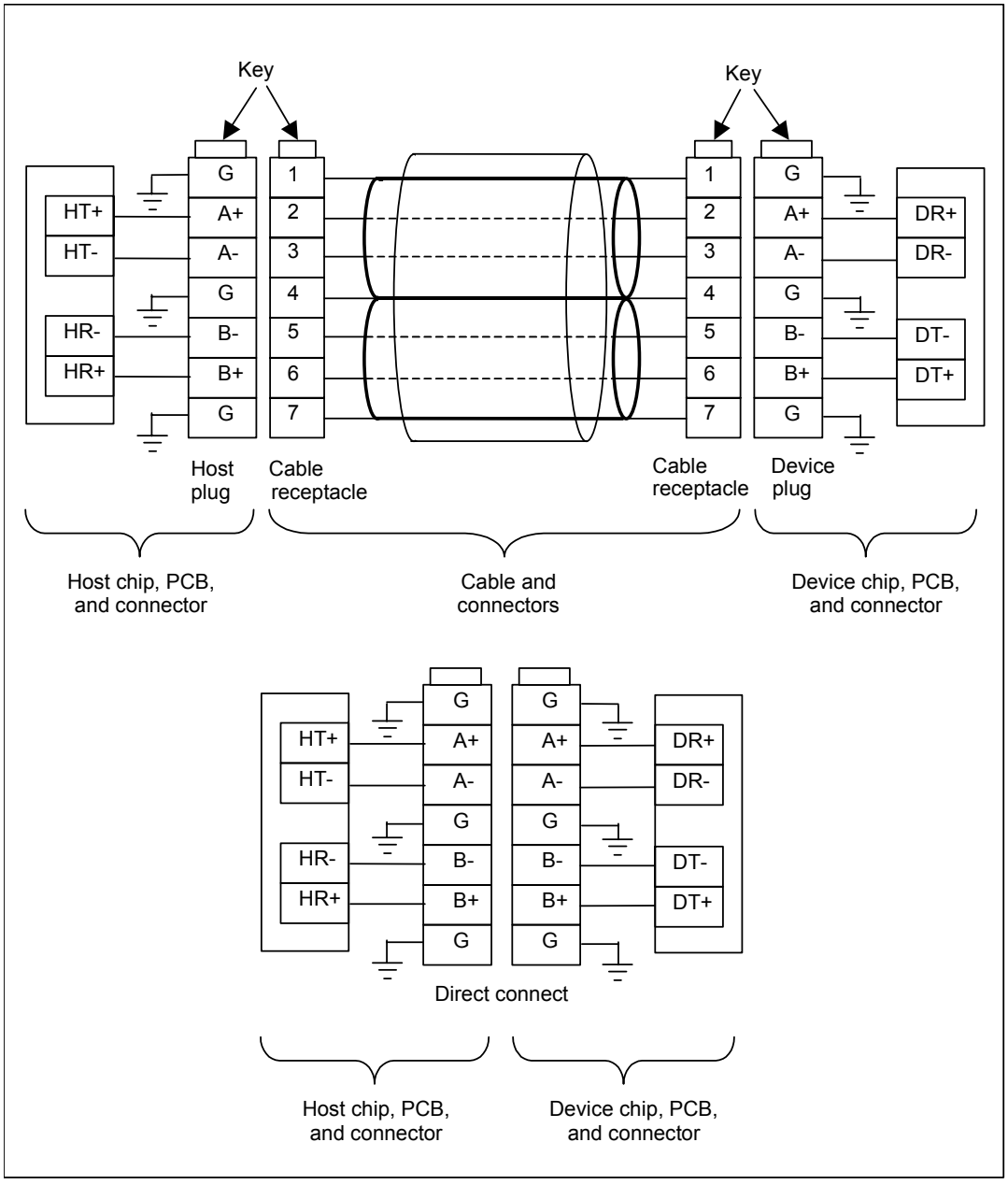


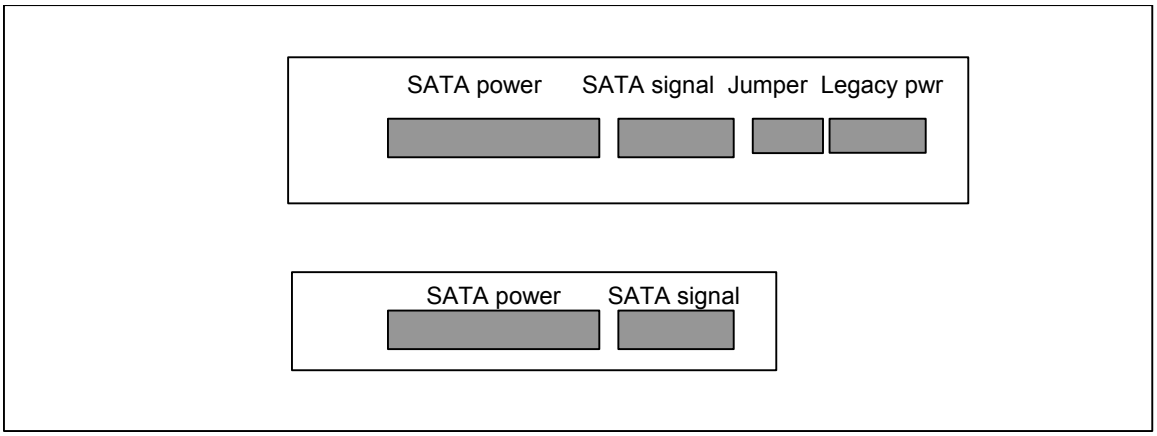
Figure 6 - Signals and grounds assigned in direct connect and cabled configurations

Illustrated above: G: ground; A+/A- and B+/B-: connector differential pairs; HT+/HT-: host transmitter signal pair; HR+/HR-: host receiver signal pair; DT+/DT-: device transmitter signal pair; DR+/DR-: device receiver signal pair.

6.3.4 Connector configurations and locations

6.3.4.1 Configurations

Serial ATA defines several connector configurations on devices. The figures below illustrate the reference connector configurations for 3.5” and 2.5” form factor devices. Refer to Appendix C for all the defined connector configurations on devices.



The reference connector configuration for the 3.5” device includes the Serial ATA connector with both signal and power segments, and the legacy jumper and power connectors for smooth transition from Parallel ATA.

The reference configuration for the 2.5” device is the only configuration allowed for the 2.5" form factor.

6.3.4.2 Locations

The device connector location is defined to facilitate blind mating.

Figure 7 and Figure 8 define the connector locations on 3.5” and 2.5” devices, respectively.

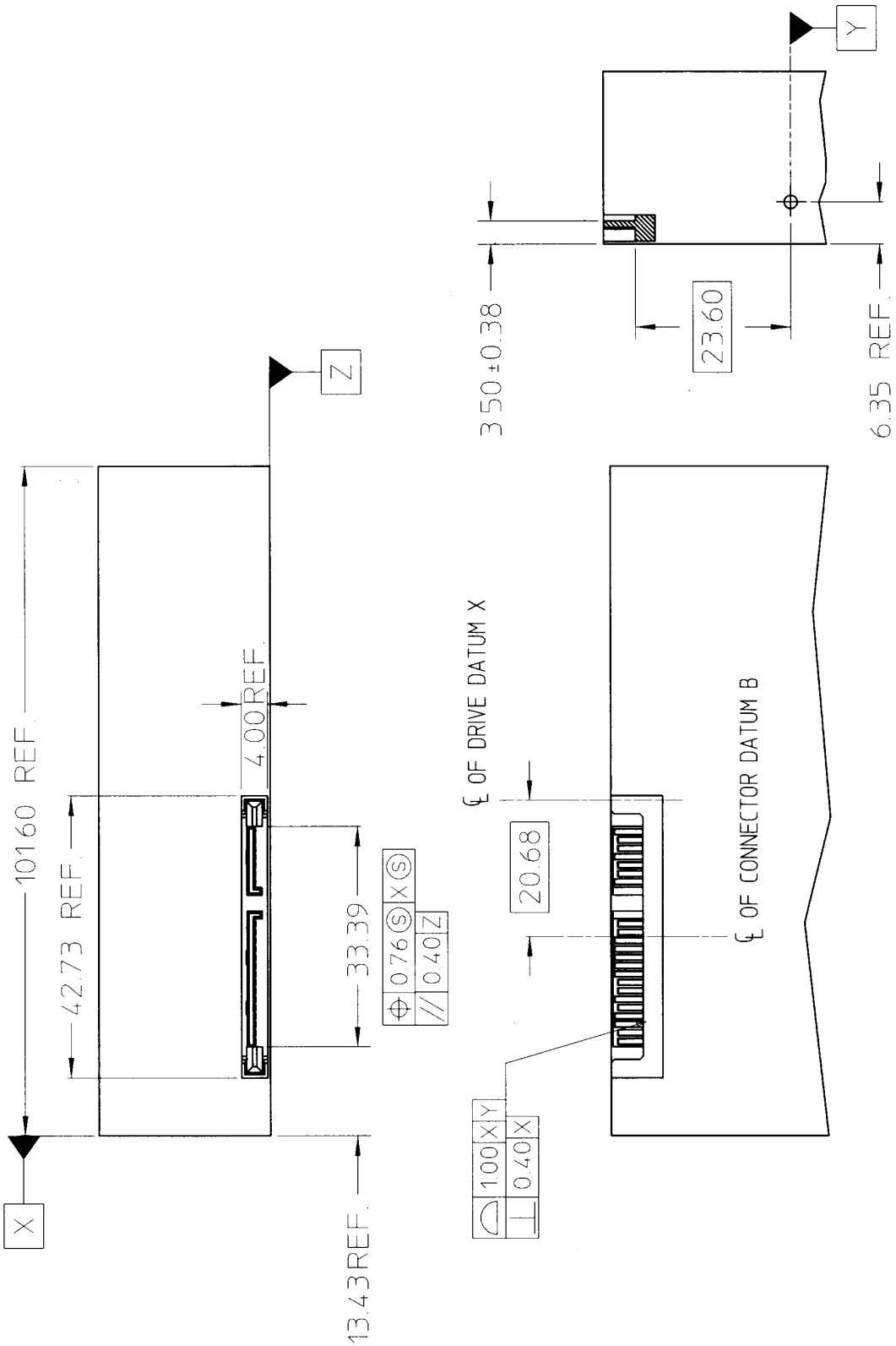


Figure 7 – Device plug connector location on 3.5” device

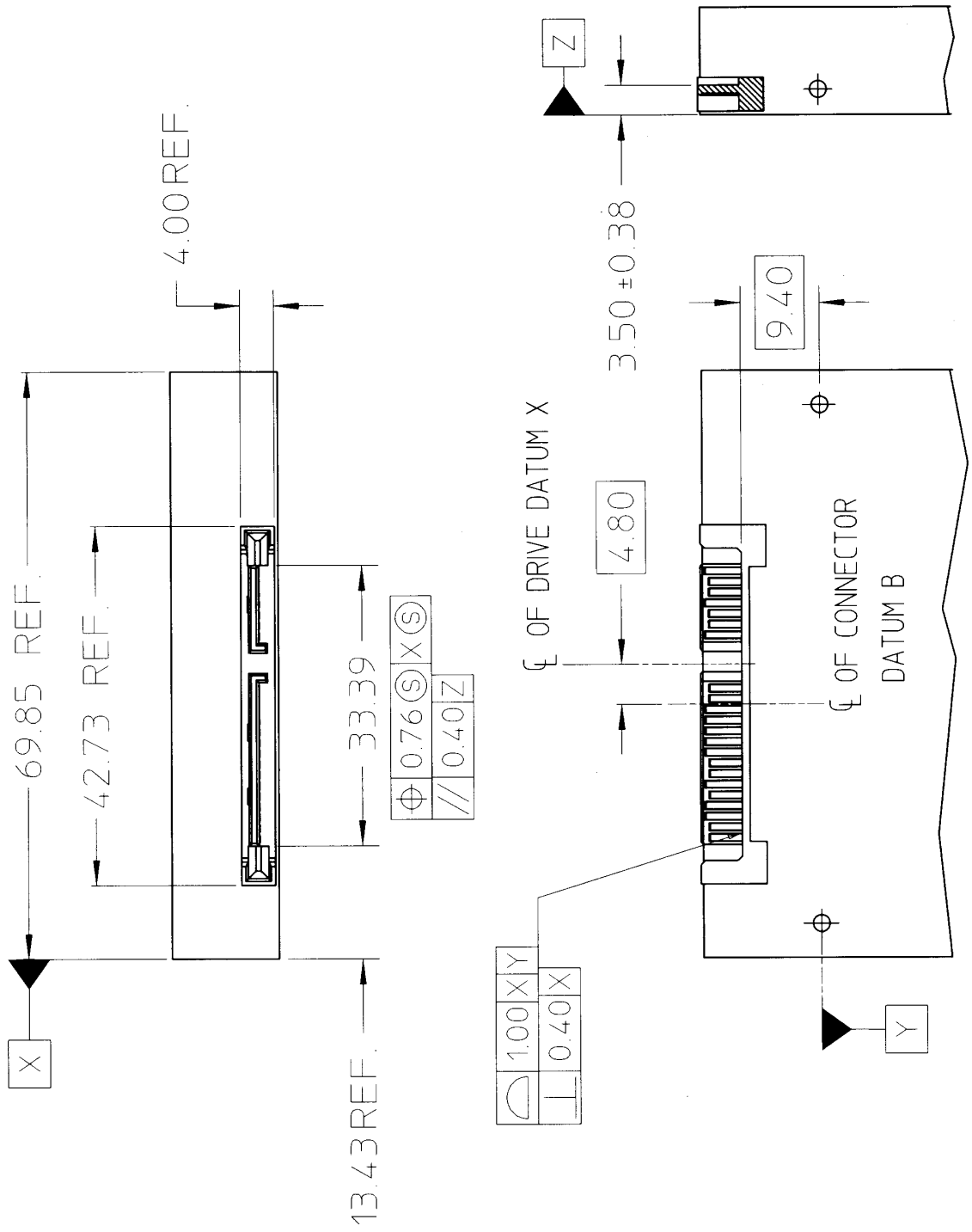


Figure 8 – Device plug connector location on 2.5” device

6.3.5 Mating interfaces

Serial ATA connectors are defined in terms of their mating interface or front end characteristics only. Connector back end characteristics including PCB mounting features and cable termination features are not defined.

6.3.5.1 Device plug connector

Figure 9 and Table 3 show the interface dimensions for the device plug connector with both signal and power segments.

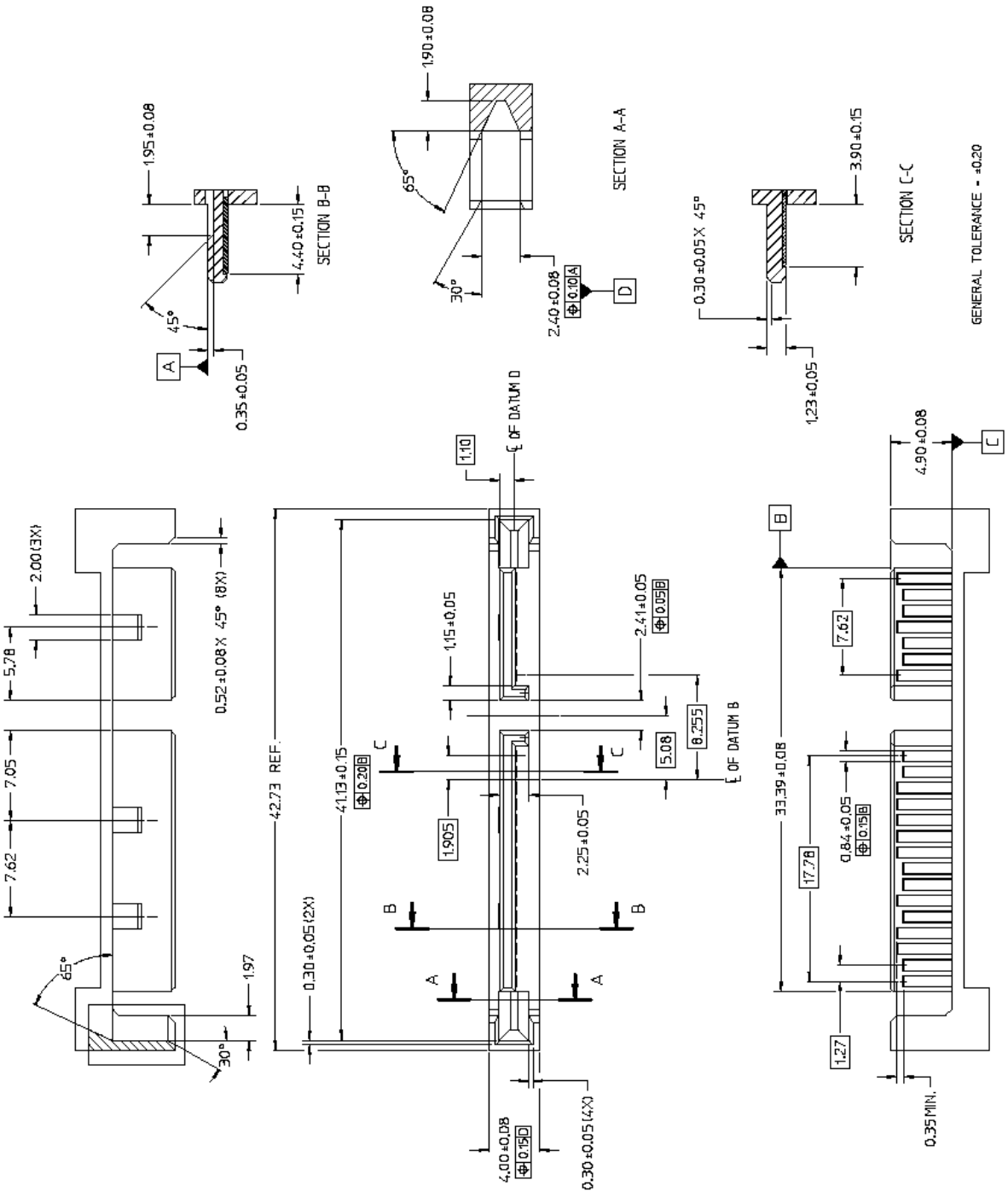


Figure 9 – Device plug connector interface dimensions

There are total of 7 pins in the signal segment and 15 pins in the power segment. The pin definitions are shown in Table 3. Note that the pin is numbered from the pin furthest from the power segment.

Table 3 – Device plug connector pin definition

Signal Segment Key			
Signal segment	S1	Gnd	2nd mate
	S2	A+	Differential signal pair A from Phy
	S3	A-	
	S4	Gnd	2nd mate
	S5	B-	Differential signal pair B from Phy
	S6	B+	
	S7	Gnd	2nd mate
Signal Segment "L"			
Central Connector Polarizer			
Power Segment "L"			
Power segment	P1	V_{33}	3.3 V power
	P2	V_{33}	3.3 V power
	P3	V_{33}	3.3 V power, pre-charge, 2nd mate
	P4	Gnd	1st mate
	P5	Gnd	2nd mate
	P6	Gnd	2nd mate
	P7	V_5	5 V power, pre-charge, 2nd mate
	P8	V_5	5 V power
	P9	V_5	5 V power
	P10	Gnd	2nd mate
	P11	Reserved	1. The pin corresponding to P11 in the backplane receptacle connector is also reserved 2. The corresponding pin to be mated with P11 in the power cable receptacle connector shall always be grounded
	P12	Gnd	1st mate
	P13	V_{12}	12 V power, pre-charge, 2nd mate
	P14	V_{12}	12 V power
	P15	V_{12}	12 V power
Power Segment Key			

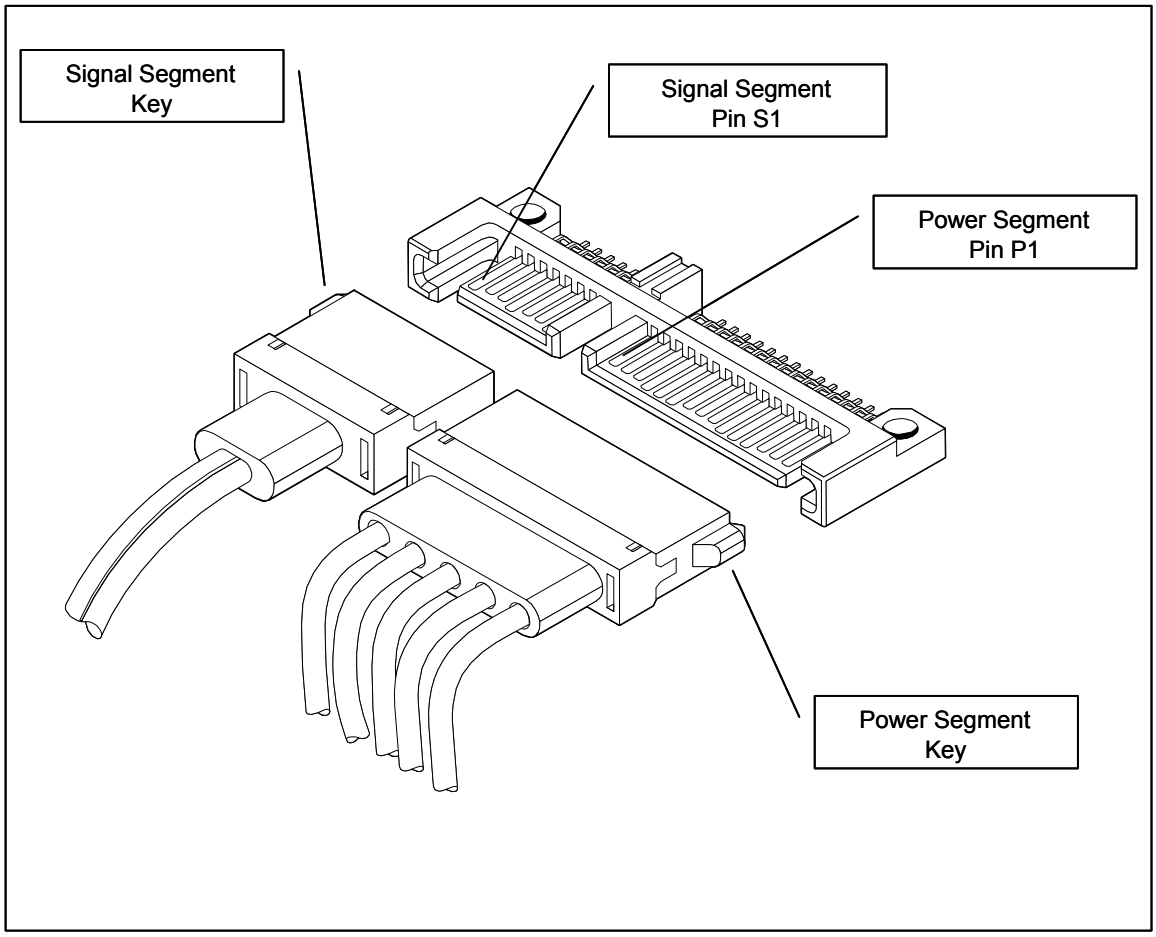


Figure 10 Connector Pin and Feature Locations

The following points should be noted:

- All pins are in a single row, with a 1.27 mm (.050”) pitch.
- The comments on the mating sequence in Table 3 apply to the case of backplane blind-mate connector only. In this case, the mating sequences are: (1) the ground pins P4 and P12; (2) the pre-charge power pins and the other ground pins; and (3) the signal pins and the rest of the power pins.
- There are three power pins for each voltage. One pin from each voltage is used for pre-charge in the backplane blind-mate situation.
- If a device uses 3.3 V, then all V_{33} pins must be terminated. Otherwise, it is optional to terminate any of the V_{33} pins.
- If a device uses 5.0 V, then all V_5 pins must be terminated. Otherwise, it is optional to terminate any of the V_5 pins.
- If a device uses 12.0 V, then all V_{12} pins must be terminated. Otherwise, it is optional to terminate any of the V_{12} pins.

6.3.5.2 Signal cable receptacle connector

Figure 11 shows the interface dimensions for the signal cable receptacle connector. There are two identical receptacles at the two ends of the Serial ATA cable assembly. The cable receptacle mates with either the signal segment of the device plug connector on the device, or the host plug connector on the host.

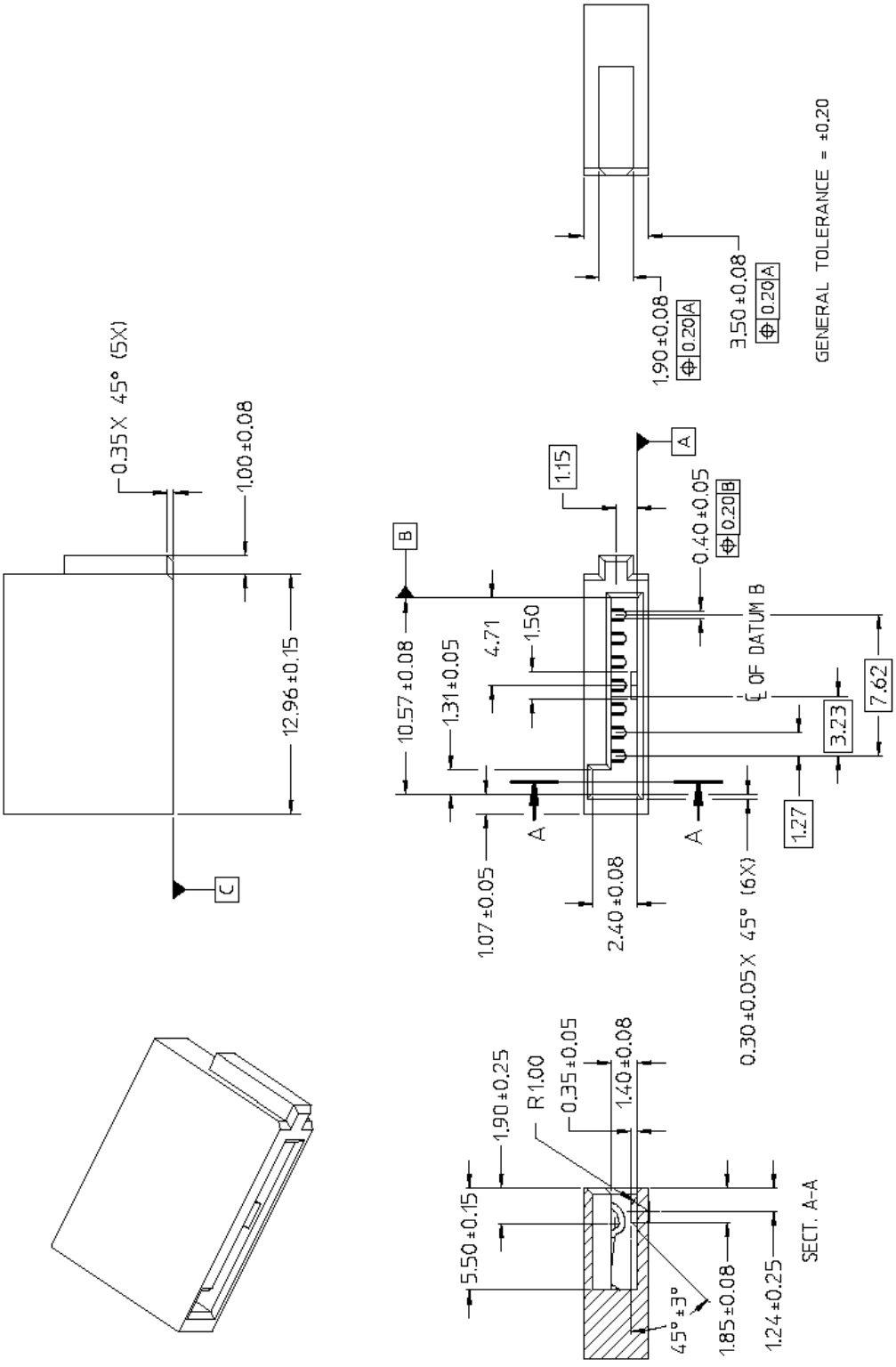


Figure 11 – Cable receptacle connector interface dimensions

The pin out of the cable receptacle connector is the mirror image of the signal segment of the device plug connector. Notice that

- The two differential pin pairs are terminated with the corresponding differential cable pairs
- The ground pins are terminated with the cable drain wires, if it applies
- The choice of cable termination methods, such as crimping or soldering is up to each connector vendor

6.3.5.3 Signal host plug connector

The signal host plug connector is to be mated with one end of the Serial ATA cable assembly. So the pinout of the host plug connector is the mirror image of the signal cable receptacle. Figure 12 shows the host plug connector interface definition.

For applications where multiple Serial ATA ports or connectors are stacked together on the host, there is a clearance or spacing requirement to prevent the cable assemblies from interfering with each other. Figure 13 shows the recommended clearance or spacing.

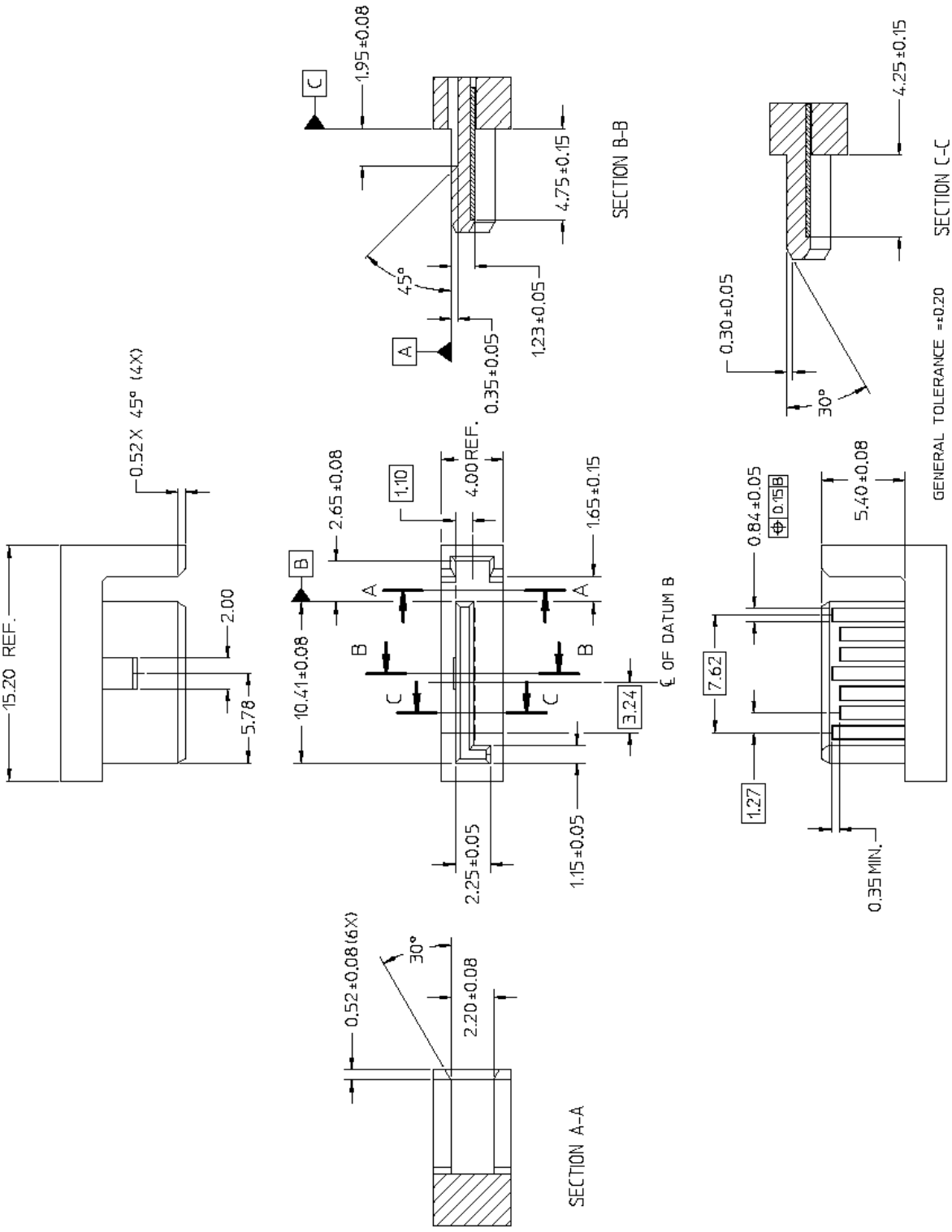


Figure 12 – Host plug connector interface dimension

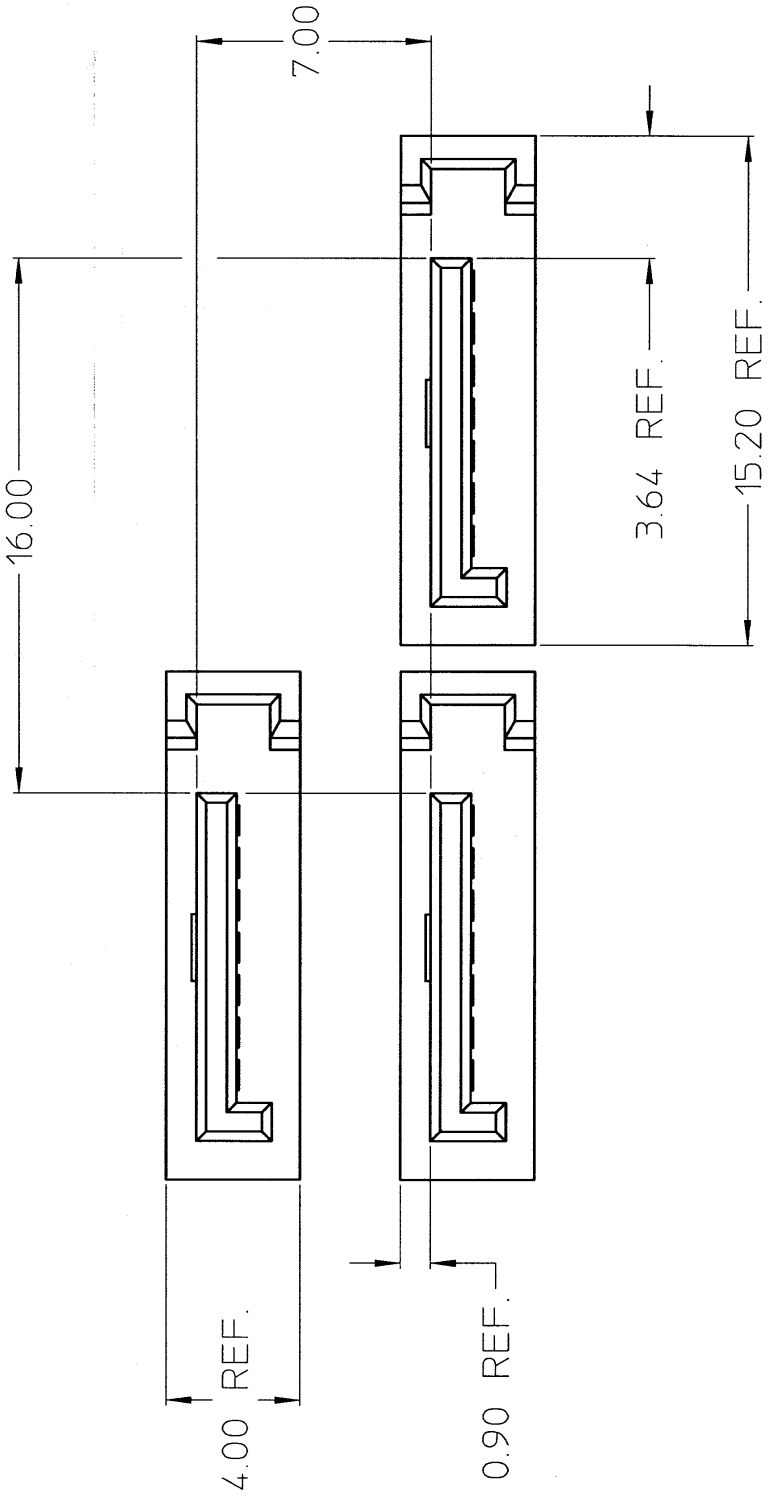


Figure 13 – Recommended host plug connector clearance or spacing

6.3.5.4 Host receptacle connector

The host receptacle connector is to be blind-mated directly with the device plug connector. The interface dimensions for the host receptacle connector are shown in Figure 14. Note that dimension B allows two values: 8.15mm and 14.15mm. There are two levels of contacts in the host receptacle connector. The advancing ground contacts P4 and P12 mate first with the corresponding ground pins on the device plug connector, followed by the engaging of the pre-charged power pins. An appropriate external retention mechanism independent of the connector is required to keep the host PCB and the device in place. The host receptacle connector is not designed with any retention mechanism.

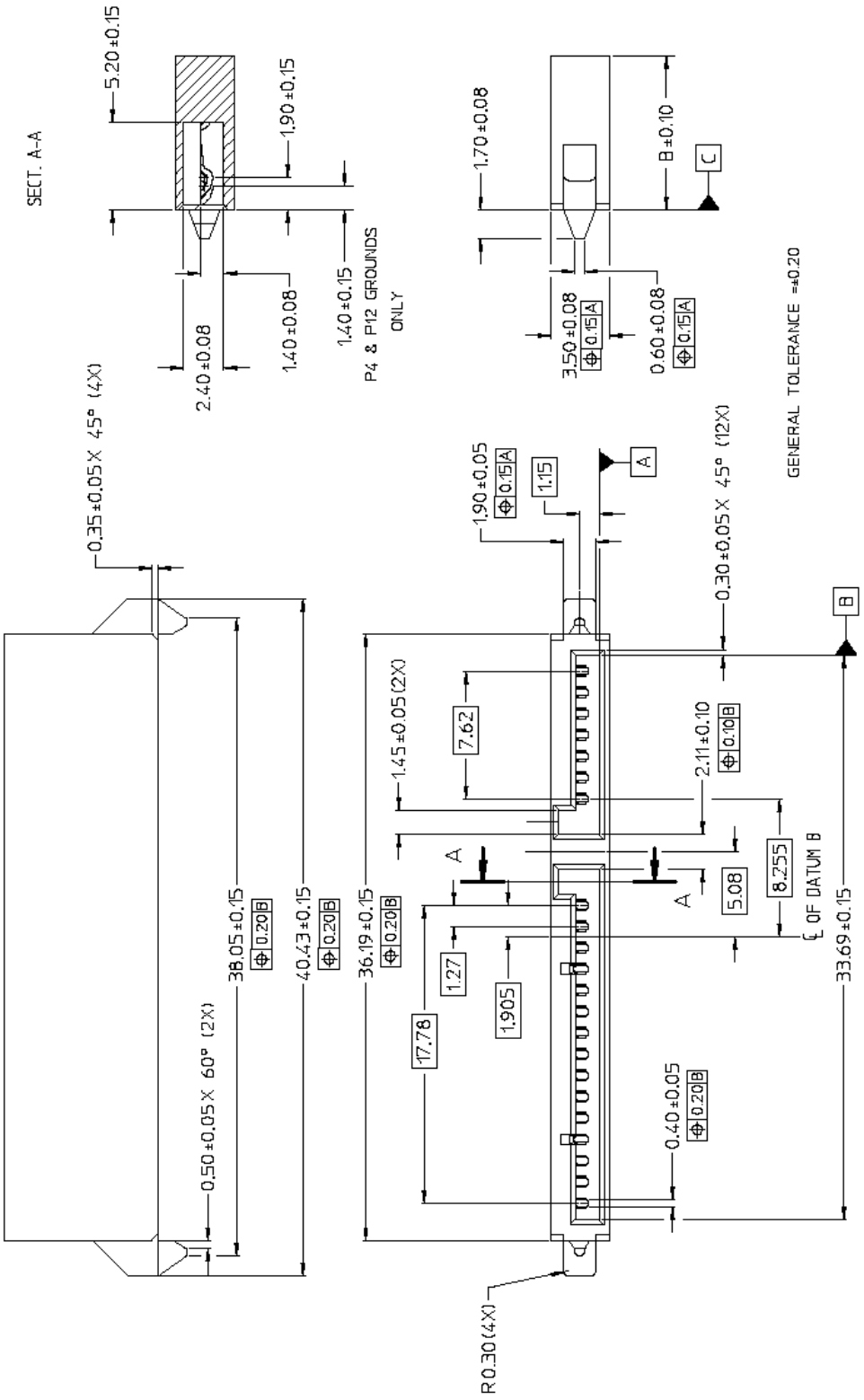


Figure 14 – Host receptacle connector interface dimensions

6.3.5.5 Power cable receptacle connector

The power cable receptacle connector mates with the power segment of the device plug, bringing power to the device. Figure 15 shows the interface dimensions of the power receptacle connector. The pinout of the connector is the mirror image of the power segment of the device plug shown in Table 3.

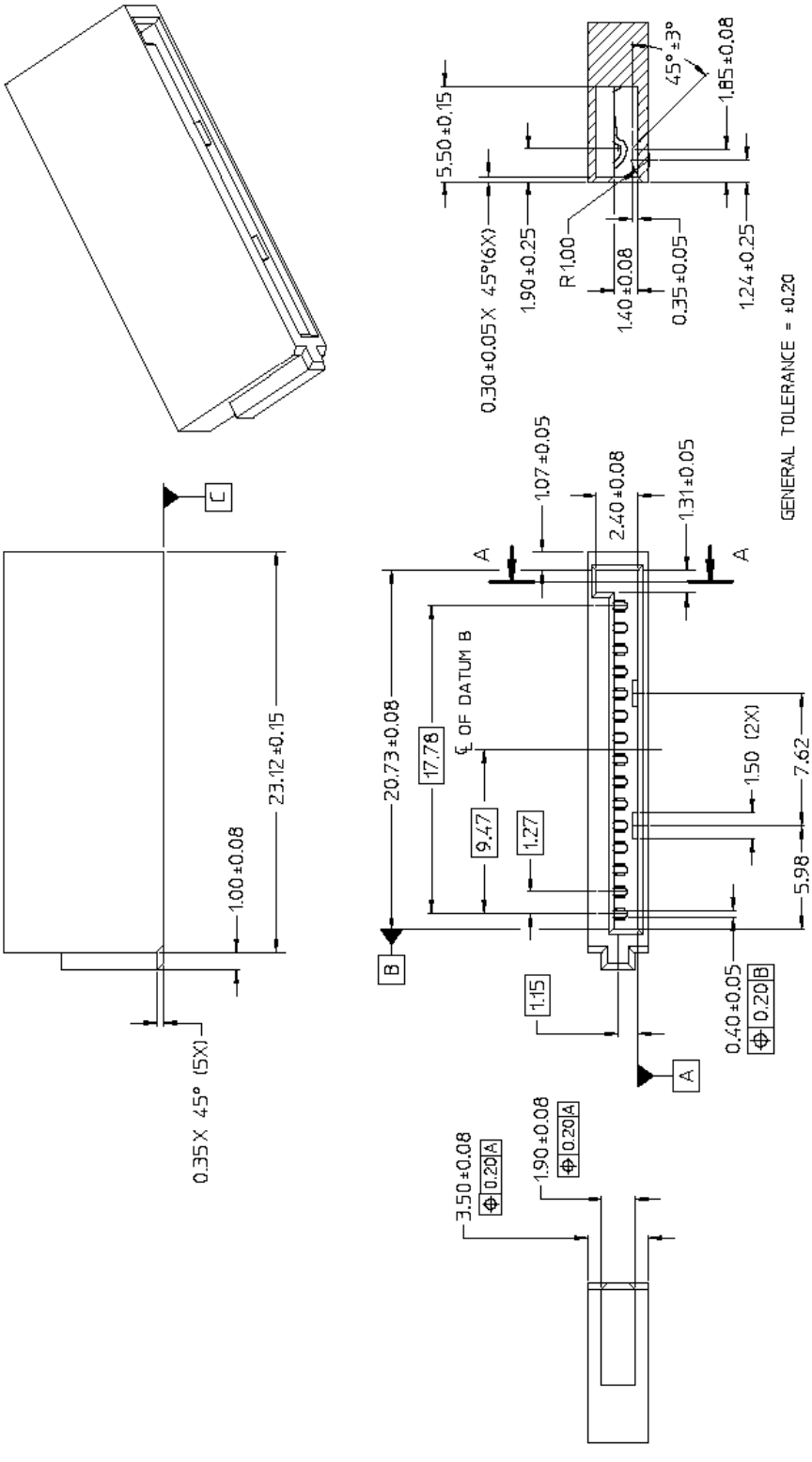


Figure 15 – Power receptacle connector interface dimensions

The power receptacle connector is terminated onto 18 AWG wires that are connected to the system power supply or other power sources. Five 18 AWG wires may be used, with three wires terminated to the nine power pins for the three voltages, while the remaining two wires to the six ground pins.

6.3.6 Serial ATA cable

The Serial ATA cable consists of four conductors in two differential pairs. If necessary, the cable may also include drain wires to be terminated to the ground pins in the Serial ATA cable receptacle connectors. The cable size may be 30 to 26 AWG. The cable maximum length is one meter.

This specification does not specify a standard Serial ATA cable. Any cable that meets the electrical requirements to be described later in this section is considered an acceptable Serial ATA cable. The connector and cable vendors have the flexibility to choose cable constructions and termination methods based on performance and cost considerations. An example cable construction is given in Appendix J.2 for an informational purpose only.

6.3.7 Backplane connector configuration and blind-mating tolerance

The maximum blind-mate misalignment tolerances are ±1.50 mm and ±1.00 mm, respectively, for two perpendicular axes illustrated in Figure 16. Any skew angle of the plug, with respect to the receptacle, will reduce the blind-mate tolerances.

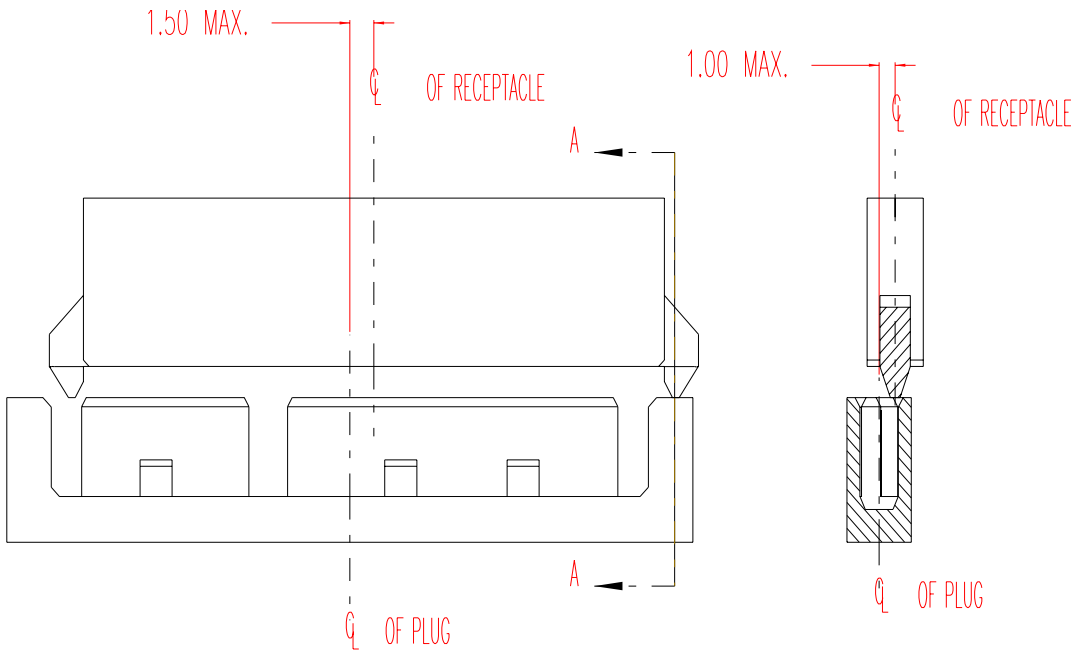


Figure 16 – Connector pair blind-mate misalignment tolerance

The device-to-backplane mating configuration is shown in Figure 17 - Device-backplane mating configuration. Note that two values (8.45 and 14.45 mm) are allowed for dimension A

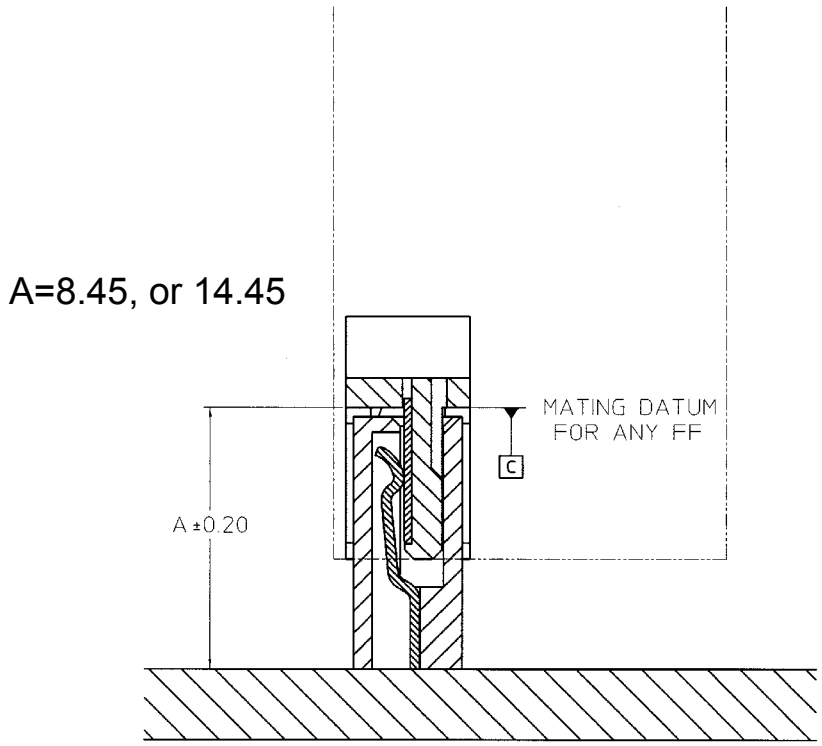


Figure 17 - Device-backplane mating configuration

6.3.8 Connector labeling

Labeling on a connector with the connector manufacturing logo or Serial ATA logo is optional.

6.3.9 Connector and cable assembly requirements and test procedures

Unless otherwise specified, all measurements shall be performed within the following lab conditions:

- Mated
- Temperature: 15° to 35° C
- Relative Humidity: 20% to 80%
- Atmospheric Pressure: 650 mm to 800 mm of Hg

If an EIA (Electronic Industry Association) test is specified without a letter suffix in the test procedures, the latest approved version of that test shall be used.

6.3.9.1 Signal

The test board shall consist of differential traces (100 ohm+-5 ohm) over a ground plane (single ended 50 ohm +/-2.5 ohm).

Open or shorted traces with the same length as the input signal traces shall be provided to measure the system input risetime and to synchronize pulses. Traces for crosstalk measurements will diverge from each other. Provisions for attenuation reference measurement shall also be provided.

Unless otherwise specified, the requirements are for the entire signal path from the host connector mated pair to the device plug mated pair connector, but not including PCB traces.

A cable assembly shall meet Table 4 electrical signaling parameters and requirements when tested with the above specified test fixture or equivalent.

Table 4 – Signal integrity requirements and test procedures

Parameter	Procedure	Requirements
Mated connector impedance	1. Minimize skew (see NOTE 1). 2. Set the Time Domain Reflectometer (TDR) pulsers in differential mode with a positive going pulse (V+) and a negative going pulse (V-). Define a reflected differential trace: Vdiff = V+ - V-. 3. With the TDR connected to the risetime reference trace, verify an input risetime of 70 ps (measured 20%-80% Vp). Filtering may be used to slow the system down (see NOTE 2). 4. Connect the TDR to the sample measurement traces. Calibrate the instrument and system (see NOTE 3). 5. Measure and record the maximum and minimum values of the near end connector impedance.	100 ohm +/- 15%
Cable absolute impedance	1. Minimize skew (see NOTE 1). 2. Set the Time Domain Reflectometer (TDR) pulsers in differential mode with a positive going pulse (V+) and a negative going pulse (V-). Define a reflected differential trace: Vdiff = V+ - V-. 3. With the TDR connected to the risetime reference trace verify an input risetime of 70 ps (measured 20%-80% Vp). Filtering may be used to slow the system down (see NOTE 2). 4. Connect the TDR to the sample measurement traces. Calibrate the instrument (see NOTE 3). 5. Measure and record maximum and minimum cable impedance values in the first 500 ps of cable response following any vestige of the connector response.	100 ohm +/- 10%

Parameter	Procedure	Requirements
Cable pair matching	1. Set the Time Domain Reflectometer (TDR) to differential mode. 2. With the TDR connected to the risetime reference traces verify an input risetime of 70 ps (measured 20%-80% Vp). Filtering may be used to slow the system down (see NOTE 2). 3. Connect the TDR to the sample measurement traces. Calibrate the instrument and system (see NOTE 3). 4. Measure and record the single–ended cable impedance of each cable within a pair Measure and record maximum and minimum cable impedance values in the first 500 ps of cable response following any vestige of the connector response. 5. The parameter then equals $Line1_{imp}$ - $Line2_{imp}$.	+/- 5 ohm
Common mode impedance	1. Set two TDR pulsers to produce a differential signal. 2. Minimize skew (see NOTE 1). 3. With the TDR connected to the risetime reference trace verify an input risetime of 70 ps (measured 20-80%) Filtering may be used to slow the system down (see NOTE 2). 4. Calibrate the TDR (see NOTE 3). 5. Set both TDR pulsers to produce positive going pulses. 6. Measure the even mode impedance of the first pulser. Divide this by 2 to get the common mode impedance. 7. Do the same for the other pulser. Both values shall meet the requirement.	25 to 40 ohms

Parameter	Procedure	Requirements
Insertion loss	1. Produce a differential signal with the signal source (see NOTE 4). 2. Assure that skew between the pairs is minimized. (see NOTE 1). 3. Measure and store the insertion loss (IL) of the fixturing, using the IL reference traces provided on the board, over a frequency range of 10 to 4500 MHz. 4. Measure and record the IL of the sample, which includes fixturing IL, over a frequency range of 10 to 4500 MHz. 5. The IL of the sample is then the results of procedure 4 minus the results of Procedure 3.	6 dB max
Crosstalk: NEXT	1. Produce a differential signal with the signal source (see NOTE 1). 2. Connect the source to the risetime reference traces. Assure that skew between the pairs is minimized. (see NOTE 1). 3. Terminate the far ends of the reference trace with loads of characteristic impedance. 4. Measure and record the system and fixturing crosstalk. This is the noise floor. 5. Terminate the far ends of the drive and listen lines with loads of characteristic impedance. 6. Connect the source to the drive pair and the receiver to the near-end of the listen pair. 7. Measure the NEXT over a frequency range of 10 to 4500 MHz. 8. Verify that the sample crosstalk is out of the noise floor.	-26 dB

Parameter	Procedure	Requirements
Rise time	1. Minimize skew (see NOTE 1). 2. Set the Time Domain Reflectometer (TDR) pulsers in differential mode with a positive going pulse (V+) and a negative going pulse (V-). Define a reflected differential trace on the receive channels as: Vdiff = V + -V-. 3. With the TDR connected to the risetime reference trace measure and record the input risetime. Verify that the input risetime is between 25-35 ps (measured 20%-80% Vp) see NOTE 2. 4. Remove the reflected trace definition. 5. Connect the TDR to the sample measurement traces. 6. Define a differential trace on the receive channels as: Vdiff = V + -V-. 7. Measure (measured 20%-80% Vp) and record the output risetime.	85 ps
Inter-symbol Interference	K – 28.5 signal source running at 1.5 Gb/sec. The average position of zero crossing should not move more than specified value.	50 ps maximum

Parameter	Procedure	Requirements
Intra-Pair Skew	1. Set one of the Time Domain Reflectometer (TDR) pulsers in differential mode with a positive going pulse (V+) and a negative going pulse (V-). 2. With the TDR connected to the risetime reference trace verify an input risetime of 70 ps (measured 20%-80% Vp). Filtering may be used to slow the system down (see NOTE 2). 3. Measure propagation delay (50% of Vp) of each line in a pair single-endedly. The skew equals the difference between each single ended propagation delay.	10 ps max
NOTES – 1: Skew must be minimized. Time domain measurement equipment allows for delay adjustment of the pulses so launch times can be synchronized. Frequency domain equipment will require the use of phase matched fixturing. The fixturing skew should be verified to be <1 ps on a TDR. 2: The system risetime is to be set via equipment filtering techniques. The filter risetime is significantly close to the stimulus risetime. Therefore the filter programmed equals the square root of ($t_{r(observed)}$)squared – ($t_{r(stimulus)}$)squared. After filtering, verify the risetime is achieved using the risetime reference traces on the PCB fixture. 3: Calibrate the system by substituting either precision 50 ohm loads or precision air lines (also terminated in 50 ohm loads) for the test fixture. This places the calibration plane directly at the input interface of the test fixture. 4: A network analyzer is preferred. If greater dynamic range is required a signal generator/spectrum analyzer may be used. Differential measurements require the use of a four port network analyzer although baluns or hybrid couplers may be used.		

6.3.9.2 Housing and contact electrical requirements

Table 5 is the connector housing and contact electrical requirements.

Table 5 – Housing and contact electrical parameters, test procedures, and requirements

Parameter	Procedure	Requirement
Insulation resistance	EIA 364-21 After 500 VDC for 1 minute, measure the insulation resistance between the adjacent contacts of mated and unmated connector assemblies.	1000 MΩ minimum
Dielectric withstanding voltage	EIA 364-20 Method B Test between adjacent contacts of mated and unmated connector assemblies.	The dielectric shall withstand 500 VAC for 1 minute at sea level.
Low level contact resistance (LLCR)	EIA 364-23 Subject mated contacts assembled in housing to 20 mV maximum open circuit at 100 mA maximum	• Initially 30 mΩ maximum. • Resistance increase 15 mΩ maximum after stress
Contact current rating (Power segment)	• Mount the connector to a test PCB • Wire power pins P1, P2, P8, and P9 in parallel for power • Wire ground pins P4, P5, P6, P10, and P12 in parallel for return • Supply 6 A total DC current to the power pins in parallel, returning from the parallel ground pins (P4, P5, P6, P10, and P12) • Record temperature rise when thermal equilibrium is reached	1.5 A per pin minimum. The temperature rise above ambient shall not exceed 30° C at any point in the connector when contact positions are powered. The ambient condition is still air at 25° C.

6.3.9.3 Mechanical and environmental requirements

Table 6 lists the mechanical parameters and requirements, while Table 7 the environmental and reliability tests and requirements:

Table 6 – Mechanical test procedures, and requirements

Test description	Procedure	Requirement
Visual and dimensional inspections	EIA 364-18 Visual, dimensional and functional per applicable quality inspection plan.	Meets product drawing requirements.
Cable pull-out	EIA 364-38 Condition A Subject a Serial ATA cable assembly to a 40 N axial load for a min of one minute while clamping one end of the cable plug.	No physical damage
Cable flexing	For round cable: EIA 364-41 Condition I Dimension x=3.7 x cable diameter, 100 cycles in each of two planes. For flat cable: EIA 364-41 Condition II 250 cycles using either Method 1 or 2.	No physical damage. No discontinuity over 1 μs during flexing.
Insertion force	EIA 364-13 Measure the force necessary to mate the connector assemblies at a max. rate of 12.5 mm (0.492") per minute.	45 N maximum
Removal force	EIA 364-13 Measure the force necessary to unmate the connector assemblies at maximum rate of 12.5 mm (0.492") per minute.	10 N minimum
Durability	EIA 364-09 50 cycles for internal cabled application; 500 cycles for backplane/blindmate application. Test done at a maximum rate of 200 cycles per hour.	No physical damage. Meet requirements of additional tests as specified in the test sequence in section 6.3.9.5

Table 7 – Environmental parameters, test procedures, and requirements

Parameter	Procedure	Requirement
Physical shock	EIA 364-27 Condition H Subject mated connectors to 30 g's half-sine shock pulses of 11 msec duration. Three shocks in each direction applied along three mutually perpendicular planes for a total of 18 shocks. See NOTE 2.	No discontinuities of 1 μs or longer duration. No physical damage.
Random vibration	EIA 364-28 Condition V Test letter A Subject mated connectors to 5.35 g's RMS. 30 minutes in each of three mutually perpendicular planes. See NOTE 2.	No discontinuities of 1 μs longer duration.
Humidity	EIA 364-31 Method II Test Condition A. Subject mated connectors to 96 hours at 40° C with 90% to 95% RH.	See NOTE 1
Temperature life	EIA 364-17 Test Condition III Method A. Subject mated connectors to temperature life at +85°C for 500 hours.	See NOTE 1.
Thermal shock	EIA 364-32 Test Condition I. Subject mated connectors to 10 cycles between –55° C and +85° C.	See NOTE 1.
Mixed Flowing Gas	EIA 364-65, Class 2A Half of the samples are exposed unmated for seven days, then mated for remaining seven days. Other half of the samples are mated during entire testing.	See NOTE 1.

NOTE –

1. Shall meet EIA 364-18 Visual Examination requirements, show no physical damage, and shall meet requirements of additional tests as specified in the test sequence in section 6.3.9.5.
2. Shock and vibration test fixture is to be determined by each user with connector vendors.

An additional requirement is listed in **Table 8.**

Table 8 – Additional requirement

Parameter	Procedure	Requirement
Flammability	UL 94V-0	Material certification or certificate of compliance required with each lot to satisfy the Underwriters Laboratories follow-up service requirements.

It should be pointed out that this specification does not attempt to define the connector and cable assembly reliability requirements that are considered application-specific. It is up to users and their connector suppliers to determine if additional requirements shall be added to satisfy the application

needs. For example, a user who requires a SMT connector may want to include additional requirements for SMT connector reliability.

6.3.9.4 Sample selection

Samples shall be prepared in accordance with applicable manufacturers' instructions and shall be selected at random from current production. Each test group shall provide 100 data points for a good statistical representation of the test result. For a connector with greater than 20 pins, a test group shall consist of a minimum of five connector pairs. From these connector pairs, a minimum of 20 contact pairs per mated connector shall be selected and identified. For connectors with less than 20 pins, choose the number of connectors sufficient to provide 100 data points.

6.3.9.5 Test sequence

Table 9 shows the connector test sequences for five groups of tests.

Table 9 – Connector test sequences

Test group ➔	**A**	**B**	**C**	**D**	**E**
Test or examination ⬇					
Examination of the connector(s)	1, 5	1, 9	1, 8	1, 8	1, 7
Low-Level Contact Resistance (LLCR)	2, 4	3, 7	2, 4, 6		4, 6
Insulation resistance				2, 6	
Dielectric withstanding voltage				3, 7	
Current rating			7		
Insertion force		2			
Removal force		8			
Durability	3	4[(a)]			2[(a)]
Physical shock		6			
Vibration		5			
Humidity				5	
Temperature life			3		
Reseating (manually unplug/plug three times)			5		5
Mixed Flowing Gas					3
Thermal shock				4	

NOTE –
(a) Preconditioning, 20 cycles for the 50-durability cycle requirement, 50 cycles for the 500-durability cycle requirement. The insertion and removal cycle is at the maximum rate of 200 cycles per hour.

For example, in Test Group A, one would perform the following tests:

1. Examination of the connector(s)
2. LLCR
3. Durability
4. LLCR
5. Examination of the connector(s)

6.4 Low level electronics block diagram

6.4.1 Diagram

The following block diagrams are provided as a reference for the following sections of this document. Although informative in nature, the functions of the blocks described herein provide the basis upon which the normative specifications apply. The individual blocks provided are considered the "industry standard" way to approach this design and are provided as an example of one possible implementation.

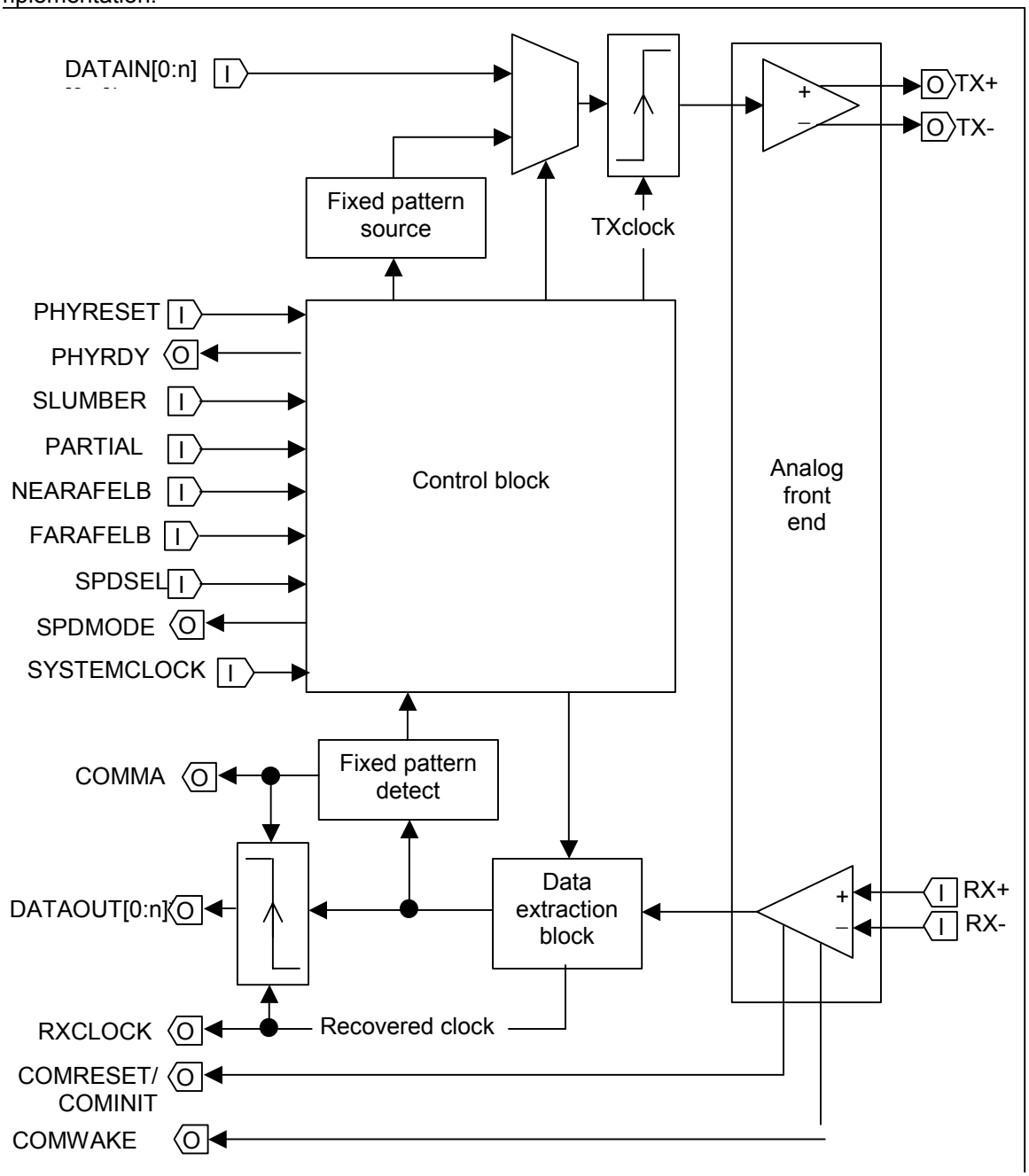


Figure 18 – Physical plant overall block diagram

6.4.2 Physical plant overall block diagram description

Analog front end	This block is the basic interface to the transmission line. This block consists of the high speed differential drivers and receivers as well as the Out of Band signaling circuitry.
Control block	This block is a collection of logic circuitry that controls the overall functionality of the Physical plant circuitry.
Fixed pattern source	This block provides the support circuitry that generates the patterns as needed to implement the ALIGN primitive activity.
Fixed pattern detect	This block provides the support circuitry to allow proper processing of the ALIGN primitives.
Data extraction block	This block provides the support circuitry to separate the clock and data from the high speed input stream.
TX clock	This signal is internal to the Physical plant and is a reference signal that regulates the frequency at which the serial stream is sent via the high speed signal path
TX + / TX -	These signals are the outbound high speed differential signals that are connected to the serial ATA cable.
RX + / RX -	These signals are the inbound high speed differential signals that are connected to the serial ATA cable.
DATAIN	Data sent from the Link layer to the Phy layer for serialization and transmission.
PHYRESET	This input signal causes the PHY to initialize to a known state and start generating the COMRESET Out of Band signal across the interface.
PHYRDY	Signal indicating Phy has successfully established communications. The Phy is maintaining synchronization with the incoming signal to its receiver and is transmitting a valid signal on its transmitter.
SLUMBER	Causes the Phy layer to transition to the Slumber power management state.
PARTIAL	Causes the Phy layer to transition to the Partial power management state
NEARAFELB	Causes the Phy to loop back the serial data stream from its transmitter to its receiver
FARAFELB	Causes the Phy to loop back the serial data stream from its receiver to its transmitter
SPDSEL	Causes the control logic to automatically negotiate for a usable interface speed or sets a particular interface speed. The actual functionality of this input is vendor specific and varies from manufacturer to manufacturer.
SPDMODE	Output signal that reflects the current interface speed setting. The actual functionality of this signal is vendor specific and varies from manufacturer to manufacturer.

SYSTEMCLOCK This input is the reference clock source for much of the control circuit and is the basis from which the transmitting interface speed is established.

COMMA This signal indicates that a K28.5 character was detected in the inbound high speed data stream.

DATAOUT Data received and deserialized by the Phy and passed to the Link layer

RX CLOCK / Recovered clock - This signal is derived from the high speed input data signal and determines when parallel data has been properly formed at the DATAOUT pins and is available for transfer to outside circuitry.

COMRESET / COMINIT
Host: Signal from the out of band detector that indicates the COMINIT out of band signal is being detected.
Device: Signal from the out of band detector that indicates the COMRESET out of band signal is being detected.

COMWAKE Signal from the out of band detector that indicates the COMWAKE out of band signal is being detected.

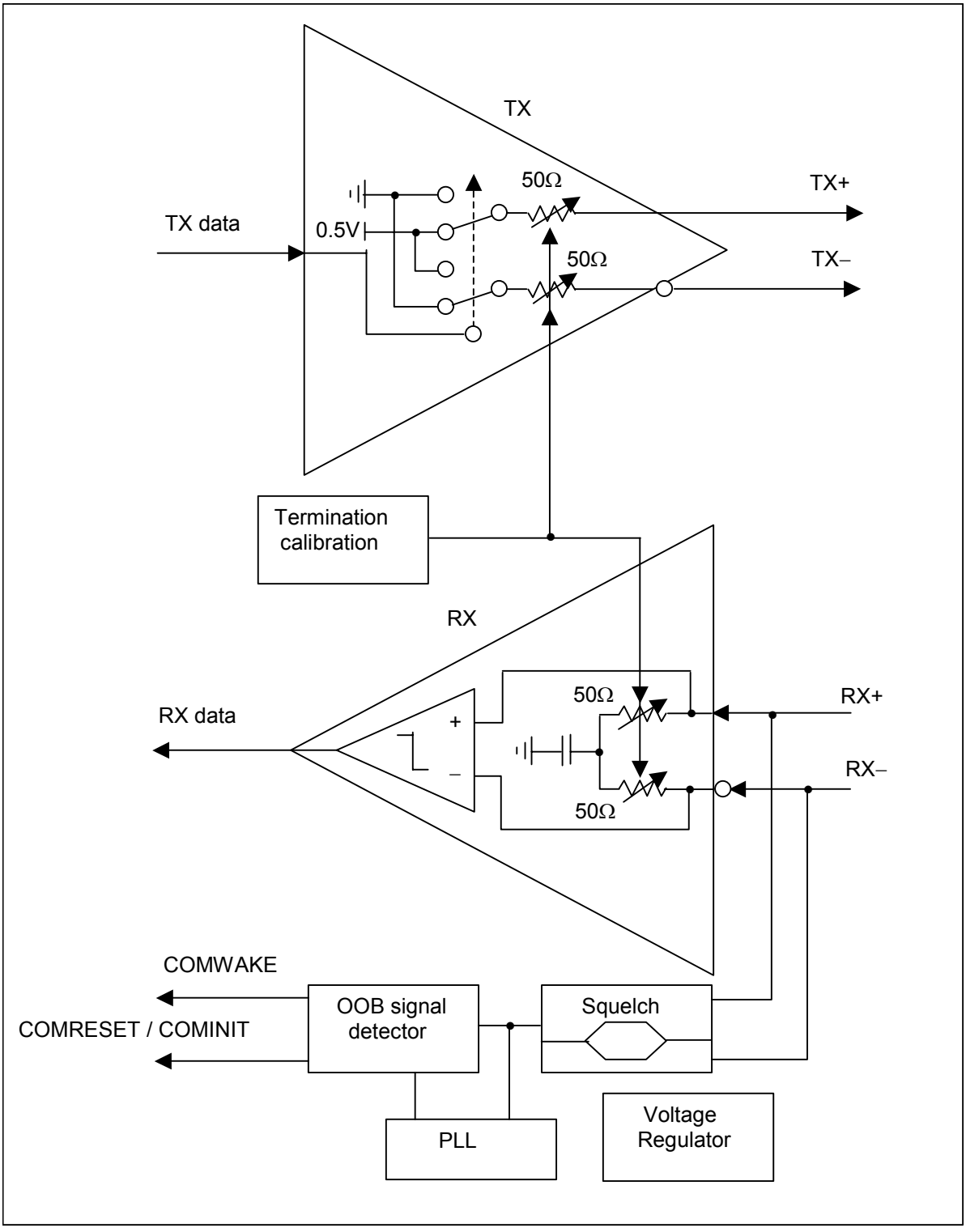


Figure 19 – Analog front end (AFE) block diagram

6.4.3 Analog front end (AFE) block diagram description

TX	This block contains the basic high speed driver electronics
RX	This block contains the basic high speed receiver electronics
Termination calibration	This block is used to establish the impedance of the RX block in order to properly terminate the high speed serial cable.
Squelch	This block establishes a limit so that detection of a common mode signal can be properly accomplished.
OOB signal detector	This block decodes Out of Band signal from the high speed input signal path.
PLL	This block is used to synchronize an internal clocking reference so that the input high speed data stream may be properly decoded.
Voltage Regulator	This block stabilizes the internal voltages used in the other blocks so that reliable operation may be achieved. This block may or may not be required for proper operation of the balance of the circuitry. The need for this block is implementation specific.
TX + / TX -	This is the same signal as described in the previous section: Physical plant overall block diagram description
RX + / RX -	This is the same signal as described in the previous section: Physical plant overall block diagram description
TxData	Serially encoded 10b data attached to the high speed serial differential line driver.
RxData	Serially encoded 10b data attached to the high speed serial differential line receiver.
COMWAKE	This is the same signal as described in the previous section: Physical plant overall block diagram description
COMRESET / COMINIT	This is the same signal as described in the previous section: Physical plant overall block diagram description

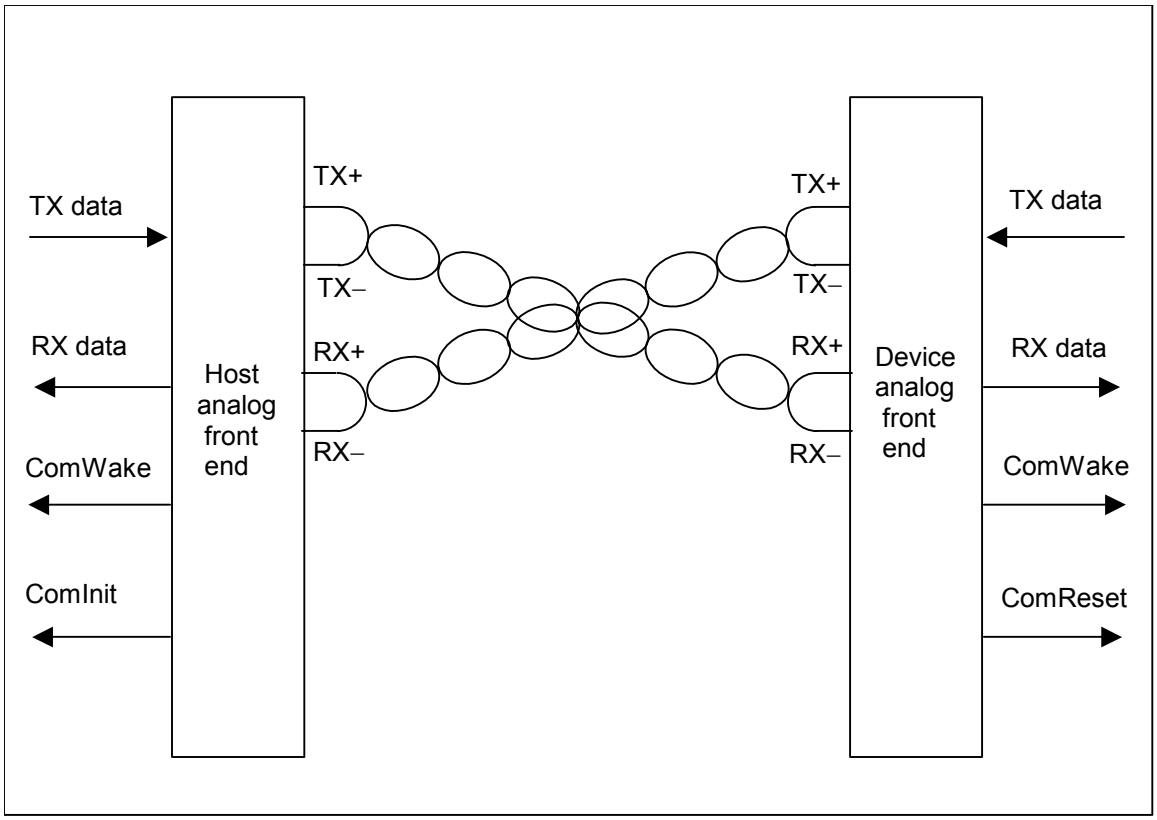


Figure 20 – Analog front end (AFE) cabling

6.5 General specifications

6.5.1 System

Table 10. – General system specifications

Name	Nom	Min	Max	Units	Description
Φ_{G1}	1.2			Gbits/s	1st Generation 8b data rate
ϕ_{G1}	1.5			Gbits/s	1st Generation 10b bit rate
Φ_{G2}	2.4			Gbits/s	2nd Generation 8b data rate
ϕ_{G2}	3.0			Gbits/s	2nd Generation 10b bit rate
Φ_{G3}	4.8			Gbits/s	3rd Generation 8b data rate
ϕ_{G3}	6.0			Gbits/s	3rd Generation 10b bit rate

6.6 Module specifications

All specifications in this section apply to a Generation 1 Serial ATA device. Generation 2 and beyond will be addressed in a separate document.

6.6.1 Definitions

RJ Random Jitter (peak to peak). Assumed to be Gaussian and equal to 14 times the 1σ rms value given the 10^{-12} BER requirement.

DJ Deterministic Jitter (peak to peak). All jitter sources that do not have tails on their probability distribution function (i.e. values outside the bounds have probability zero). Four kinds of deterministic jitter are identified: duty cycle distortion, data dependent (ISI), sinusoidal, and uncorrelated (to the data) bounded. DJ is characterized by its bounded, peak-to-peak value.

TJ Total Jitter (peak to peak). Peak to peak measured jitter including DJ and RJ.

ISI Inter-symbol interference. Data-dependent deterministic jitter caused by the time differences required for the signal to arrive at the receiver threshold when starting from different places in bit sequences (symbols). For example media attenuates the peak amplitude of the bit sequence [0,1,0,1...], more than it attenuates the peak amplitude of the bit sequence [0,0,0,0,1,1,1,1...], thus the time required to reach the receiver threshold with the [0,1,0,1...] sequence is less than required from the [0,0,0,0,1,1,1,1...]sequence. The run length of 4 produces a higher amplitude which takes more time to overcome when changing bit values and therefore produces a time difference compared to the run length of 1 bit sequence. When different run lengths are mixed in the same transmission the different bit sequences (symbols) therefore interfere with each other. ISI is expected whenever any bit sequence has frequency components that are propagated at different rates by the transmission media. This translates into high-high-frequency, data-dependent, jitter.

UI Unit Interval. Equal to the time required to transmit one bit (666.667 ps for Gen1).

DC Strictly, the non-AC component of a signal. In this specification, DC means all frequency components below f_{dc} = 100 kHz.

Differential Signal A signal derived by taking the difference between two conductors. In this spec a differential signal is comprised of a positive conductor and a negative conductor. The differential signal is the voltage on the positive conductor minus the voltage on the negative conductor (i.e. TX+ – TX-).

Burst A short pulse of data starting from and ending with the idle condition on the interface. These are used for low-level signaling when the high-speed communication channel is not established.

SSC Spead Spectrum Clocking. The technique of modulating the operating frequency of a circuit slightly to spread its radiated emissions over a range of frequencies rather than just one tone. This reduction in the maximum emission for a given frequency helps meet FCC requirements.

6.6.2 Electrical specifications

The Serial ATA physical layer shall comply to the electrical specifications depicted in Table 11. – General electrical specifications.

The physical layer module consists of the driver, receiver, PCB and mated connector pair. The physical layer module shall meet Table 11 electrical signal parameters and requirements. Unless

otherwise specified, all measurements shall be taken through the mated connector pair. The Receiver test fixture shown in Figure 46 or the Transmitter test fixture shown in Figure 45 is used. Driver and receiver designs must compensate for the effects of the path to/from the I/O connector.

Table 11. – General electrical specifications

	Nom	Min	Max	units	Comments
T,UI		666.43	670.12	ps	Operating data period (nominal value architecture specific)
t_{rise}	0.3	0.2	0.41	UI	20%-80% at transmitter
t_{fall}	0.3	0.2	0.41	UI	80%-20% at transmitter
$V_{cm,dc}$	250	200	450	mV	Common mode DC level measured at receiver connector. This spec only applies to direct-connect designs or designs that hold the common-mode level. AC coupled designs may allow the common mode to float. See $V_{cm,ac\ coupled}$ requirements.
$V_{cm,ac\ coupled\ TX}$		0	2.0	V	Open circuit DC voltage level of each signal in the TX pair at the IC side of the coupling capacitor in an AC coupled PHY. Must be met during all possible power and electrical conditions of the PHY including power off and power ramping. Transmitter common mode DC levels outside this range may be used provided that the following is met: The common mode voltage transients measured at the TX pins of the connector into an open-circuit load during all power states and transitions shall never exceed a +2.0V or –2.0V change from the CM value at the beginning of each transient. Test conditions shall include system power supply ramping at the fastest possible power ramp (up and down) for the system using such a PHY.
$V_{cm,ac\ coupled\ RX}$		0	2.0	V	Open circuit DC voltage level of each signal in the RX pair at the IC side of the coupling capacitor in an AC coupled PHY. Must be met during all possible power and electrical conditions of the PHY including power off and power ramping. Receiver common mode DC levels outside this range may be used provided that the following is met: The common mode voltage transients measured at the RX

					pins of the connector into an open-circuit load during all power states and transitions shall never exceed a +2.0V or –2.0V change from the CM value at the beginning of each transient. Test conditions shall include system power supply ramping at the fastest possible power ramp (up and down) for the system using such a PHY.
$V_{cm,ac}$			100	mV	Max sinusoidal amplitude of common mode signal measured at receiver connector
F_{CM}		2	200	MHz	All receivers must be able to tolerate sinusoidal common-mode noise components inside this frequency range with an amplitude of $V_{cm,ac}$.
$T_{settle,CM}$			10	ns	Maximum time for common-mode transients to settle to within 10% of DC value during transitions to and from the idle bus condition.
$V_{diff,tx}$	500	400	600	mV_{p-p}	+/- 250 mV differential nominal. Measured at Serial ATA connector on transmit side
$V_{diff,rx}$	400	325	600	mV_{p-p}	+/- 200 mV differential nominal. Measured at Serial ATA connector on receive side
Tx pair differential impedance	100	85	115	Ohm	As seen by a differential TDR with 100 ps (max) edge looking into connector (20%-80%). Measured with TDR in differential mode.
Rx pair differential impedance	100	85	115	Ohm	As seen by a differential TDR with 100 ps (max) edge looking into connector (20%-80%). Measured with TDR in differential mode.
Tx single-ended impedance		40		Ohm	As seen by TDR with 100 ps (max) edge looking into connector (20%-80%). TDR set to produce simultaneous positive pulses on both signals of the Tx pair. Single-ended impedance is the resulting (even mode) impedance of each signal. Both signals must meet the single ended impedance requirement. Must be met during all possible power and electrical conditions of the PHY including power off and power ramping.
Rx single-ended impedance		40		Ohm	As seen by TDR with 100 ps (max) edge looking into connector (20%-80%). TDR set to produce simultaneous

					positive pulses on both signals of the Rx pair. Single-ended impedance is the resulting (even mode) impedance of each signal. Both signals must meet the single ended impedance requirement. Must be met during all possible power and electrical conditions of the PHY including power off and power ramping.
$C_{ACcoupling}$			12	nF	Coupling capacitance value for AC coupled TX and RX pairs.
TX DC clock frequency skew		-350	+350	ppm	Specifies the allowed ppm tolerance for TX DC frequency variations around the nominal 1.500GHz. Excludes the +0/-5000ppm SSC downspread AC modulation per 6.6.4.5.
TX AC clock frequency skew		-5000	+0	ppm	Specifies the allowed ppm extremes for the SSC AC modulation, subject to the "Downspread SSC" triangular modulation (30-33kHz) profile per 6.6.4.5. Note: Total TX Frequency variation around nominal 1.500G, includes [TXDC] + [TX AC] ppm variations.
TX differential skew			20	ps	(Nominal value architecture specific)
Squelch detector threshold	100	50	200	mV_{p-p}	Minimum differential signal amplitude
COMRESET/COMINIT detector off threshold		175	525	ns	Detector shall reject all bursts with spacings outside this spec.
COMRESET/COMINIT detector on threshold	320	304	336	ns	Detector shall detect all bursts with spacings meeting this period
COMRESET/COMINIT transmit spacing	320.0	310.4	329.6	ns	As measured from 100mV differential crosspoints of last and first edges of bursts
COMWAKE detector off threshold		55	175	ns	Detector shall reject all bursts with spacings outside this spec.
COMWAKE detector on threshold	106.7	101.3	112	ns	Detector shall detect all burst spacings meeting this period
COMWAKE transmit spacing	106.7	103.5	109.9	ns	As measured from 100mV differential crosspoints from last to first edges of bursts
UI_{OOB}		646.67	686.67	ps	Operating data period during OOB burst transmission

6.6.3 Differential voltage/timing (EYE) diagram

The EYE diagram is more of a qualitative measurement than a spec. Any low frequency (trackable) modulation must be tracked by the oscilloscope to prevent measurement error caused by benign EYE closure. It is also unrealistic to try to capture sufficient edges to guarantee the 14 sigma RJ requirement in order to achieve an effective 10^{-12} BER. Nonetheless, this method is useful and easy to set up in the lab.

If t3-t1 is set equal to $DJ+6*RJ_{1sigma}$, then less than one in every 750 edges will cross the illegal region for a well designed transmitter. By controlling the number of sweeps displayed, a quick health check may be performed with minimal setup. Note that this criteria is informative, and these conditions are not sufficient to guarantee compliance, but are to be used as part of the design guidelines.

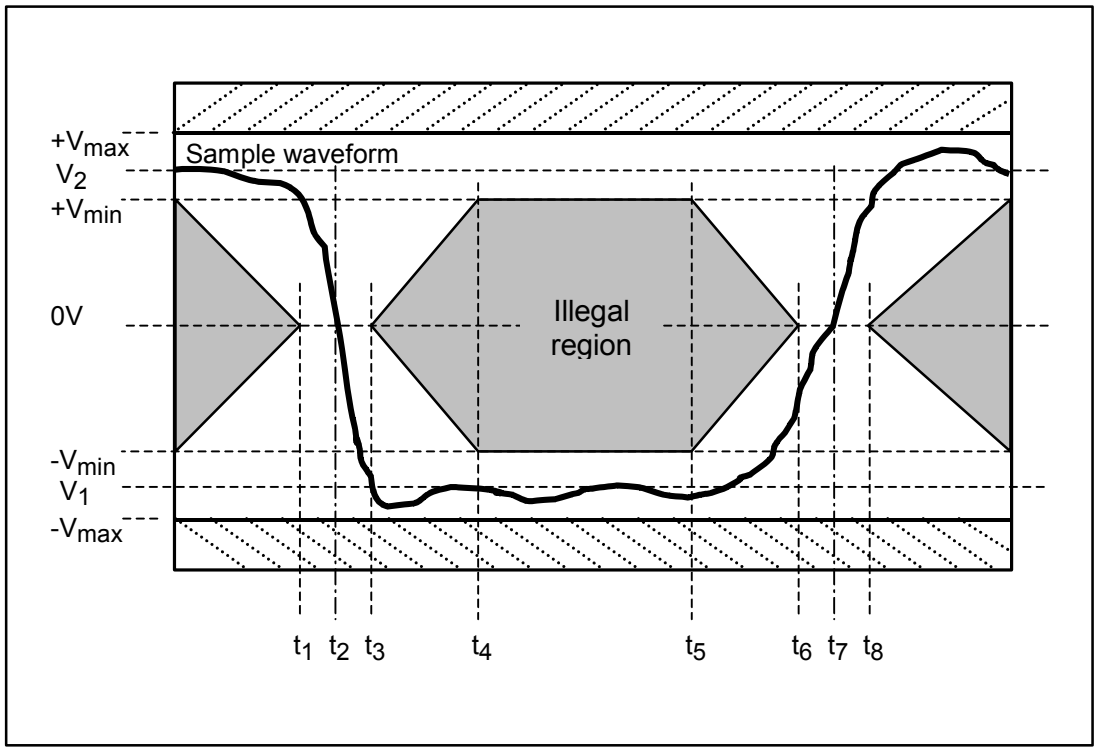


Figure 21 – Voltage / timing margin base diagram

Table 12. – Voltage / Timing Margin Definition

Name	Definition	Notes
$\mathbf{t}_{jitter}$	t3 - t1	t3 - t1 = t8 - t6
T	t7 - t2	t2 - t1 = t3 - t2 t7 - t6 = t8 - t7
$\mathbf{V}_{diff}$	$V_2 - V_1$	

6.6.4 Sampling jitter specifications

6.6.4.1 Jitter output/tolerance mask

The Serial ATA spectral jitter shall comply to the requirements as indicated in the Jitter Output/Tolerance Graph, shown in Figure 22 – Jitter output/tolerance mask with magnitudes defined in Table 13. – Sampling differential noise budget. A_x are the maximum peak to peak transmitter output and the minimum peak to peak receiver tolerance requirements as measured from a data edge to any following data edge up to n_x*UI later (where x is 0,1, or 2 in accordance with Figure 23 - Jitter as a function of frequency).

Transmitter output data edge to data edge timing variation from t_o to t_y shall not exceed the value computed by the following:

- y is an integer from 1 to n_x,
- t_y is the time between the data transition at t_0 and a data transition y*UI bit periods later.

This measurement can be made with an oscilloscope having a histogram function or with a Timing Interval Analyzer (TIA). For further information on jitter measurements see section 6.7. A receiver must be able to tolerate the peak to peak jitter specified.

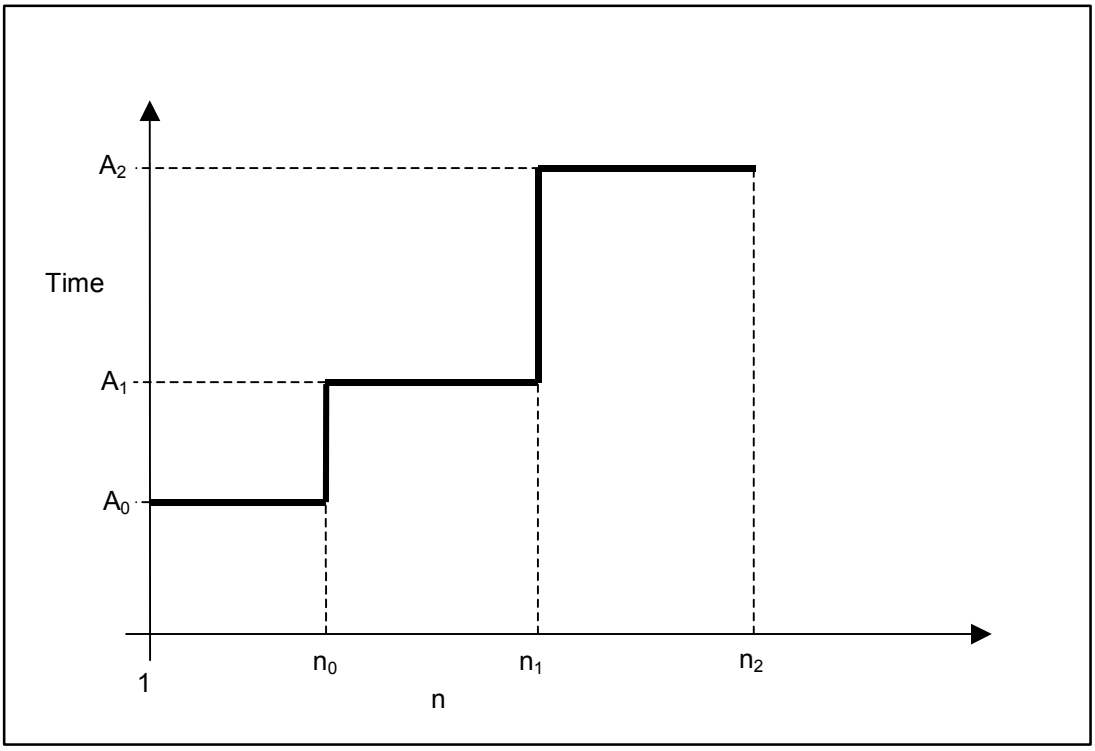


Figure 22 – Jitter output/tolerance mask

6.6.4.2 Sampling differential noise budget

Sampling jitter specifications relate to the relationship between the sampling clock and the data. Any phase error that results in the sample being improperly read (i.e. prior bit or following bit sampled) will result in a bit error.

These error components have been broken out into Deterministic Jitter (DJ) and Total Jitter (TJ) where appropriate. DJ is the peak to peak phase variation in the 0 $V_{differential}$ crossing point of the data stream that is fixed given any specific set of conditions. TJ is defined as DJ + Random Jitter (RJ). RJ is defined as 14 times the rms (1 sigma) value of the jitter that is Gaussian (normal).

The Serial ATA interface jitter characteristics should comply to within the jitter budget allocations tabulated in Table 13

Table 13. – Sampling differential noise budget

Descr	Driver output[1]		Driver PCB connector		Receiver PCB connector		Receiver input[2]		Note ref
	DJ	TJ	DJ	TJ	DJ	TJ	DJ	TJ	
$\mathbf{A_{0,p-p}}$ (UI)	0.15	0.33	0.175	0.355	0.25	0.43	0.275	0.455	4,5
$\mathbf{n_0}$	5	5	5	5	5	5	5	5	4,5
$\mathbf{A_{1,p-p}}$ (UI)	0.2	0.45	0.22	0.47	0.35	0.6	0.37	0.62	4,6
$\mathbf{n_1}$	250	250	250	250	250	250	250	250	4,6
$\mathbf{A_{2,p-p}}$ (UI)	40		40		40		40		3,4,7
$\mathbf{n_2}$	25000		25000		25000		25000		3,4,7

NOTES –
1. The driver output is the maximum jitter that a driver may exhibit to guarantee operation.
2. This field is the maximum jitter that a receiver must tolerate to guarantee operation.
3. For low frequency (trackable) jitter, total jitter is specified (DJ not broken out)
4. Does not include UI error due to frequency skew (XTAL or SSC related)
5. Primarily determined by non-tracking architecture requirements
6. Primarily determined by tracking architecture requirements
7. Primarily determined by Spread Spectrum Clocking (+/-0.25% AC portion). Doesn't include the –0.25% fixed skew (additional 26 UI).

6.6.4.3 Relationship of frequency to the jitter specification

It is often useful to look at the jitter specification as a function of frequency. This section is provided as clarifying information (informative, not normative). Figure 23 shows a plot of the maximum amplitude (in UI) sine wave at a given frequency that satisfies all the jitter specifications and the calculations leading to this graph. Two graphs are included – the second is a close-up of the high-frequency requirements.

Jitter specification at the receiver input

$A0 := 0.455 \qquad A1 := 0.62 \qquad A2 := 40$

$N0 := 5 \qquad N1 := 250 \qquad N2 := 25000$

$datarate := 1.5 \cdot 10^9$

$f(n) := \frac{datarate}{2 \cdot n}$ Frequency of sine wave with peak values separated by n datarate cycles

$a(n, n_x, a_x) := \frac{a_x}{\sin\left(\frac{\pi \cdot n_x}{2 \cdot n}\right)}$ Maximum amplitude of of sine wave with frequency f(n) and a slewrate constrained by jitter spec n_x, a_x

Maximum compliant peak-peak amplitude

$$p(n) := \begin{cases} A0 & \text{if } (n \le N0) \\ \min((a(n, N0, A0)\ A1)) & \text{if } (n > N0) \cdot (n \le N1) \\ \min((a(n, N1, A1)\ A2)) & \text{if } (n > N1) \\ 0 & \text{otherwise} \end{cases}$$

Maximum compliant peak-peak amplitude plotted against log(frequency)

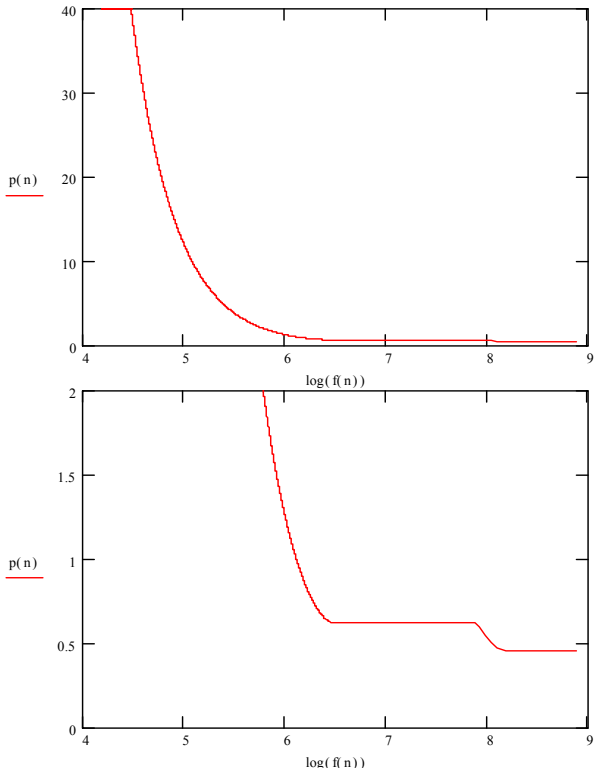


Figure 23 - Jitter as a function of frequency

6.6.4.4 Sampling BER and jitter formulas

The values for RJ and DJ above are calculated using the following relationship between BER and jitter. This is provided for reference (informative) and not intended for use as a compliance requirement (normative). The over-sampling formula assumes a worse-case oversampling ratio (osr) of three and sets the A_0,n_0 jitter requirements. The tracking architecture sets the A_1,n_1 requirement. A_2,n_2 is set to guarantee SSC compliance. Also shown below are DJ_{RX} and RJ_{RX} numbers for the two architectures. These are example budgets for the local receiver that satisfy the 10^{-12} BER goal.

Gaussian Distribution Error Function

$$G(x) := \frac{1}{\sqrt{2\cdot\pi}} \cdot \int_0^x e^{\frac{-\xi^2}{2}}\, d\xi + 0.5 \quad x > 0$$

BER Due to Sampling (oversampling architecture)

$osr := 3$ $\quad DJ_{RX} := 0.1$ $\quad RJ_{RX} := 0.1$

$DJ := 0.275 + DJ_{RX}$ $\quad RJ := 0.18 + RJ_{RX}$

$$BER_{os} := 2 - G\left[\frac{\frac{1}{osr} - \frac{DJ}{2}}{\frac{RJ}{14}}\right] - G\left[\frac{\frac{osr-1}{osr} + \frac{DJ}{2}}{\frac{RJ}{14}}\right]$$

$$BER_{os} = 1.53\cdot 10^{-13}$$

BER Due to Sampling (tracking architecture)

$DJ_{RX} := 0.15$ $\quad RJ_{RX} := 0.18$

$DJ := 0.37 + DJ_{RX}$ $\quad RJ := 0.25 + RJ_{RX}$

$$BER_{tr} := 2 - G\left[\frac{\frac{1}{2} - \frac{DJ}{2}}{\frac{RJ}{14}}\right] - G\left[\frac{\frac{1}{2} + \frac{DJ}{2}}{\frac{RJ}{14}}\right]$$

$$BER_{tr} = 2.665\cdot 10^{-15}$$

Figure 24 - Sampling bit error rate formulas

6.6.4.5 Spread spectrum clocking (SSC) clarification:

Spread Spectrum functionality define follows

1. All device timings (including jitter, skew, min-max clock period, output rise/fall time) MUST meet the existing non-spread spectrum specifications
2. The minimum clock period cannot be violated. The preferred method is to adjust the spread technique to not allow for modulation above the nominal frequency. This technique is often called “down-spreading”. An example triangular frequency modulation profile is shown in Figure 25. The modulation profile in a modulation period can be expressed as:

$$f = \begin{cases} (1-\delta) f_{nom} + 2 f_m \cdot \delta \cdot f_{nom} \cdot t & when \quad 0 < t < \dfrac{1}{2 f_m}; \\ (1+\delta) f_{nom} - 2 f_m \cdot \delta \cdot f_{nom} \cdot t & when \quad \dfrac{1}{2 f_m} < t < \dfrac{1}{f_m}, \end{cases}$$

where f_{nom} is the nominal frequency in the non-SSC mode, f_m is the modulation frequency, δ is the modulation amount, and t is time.

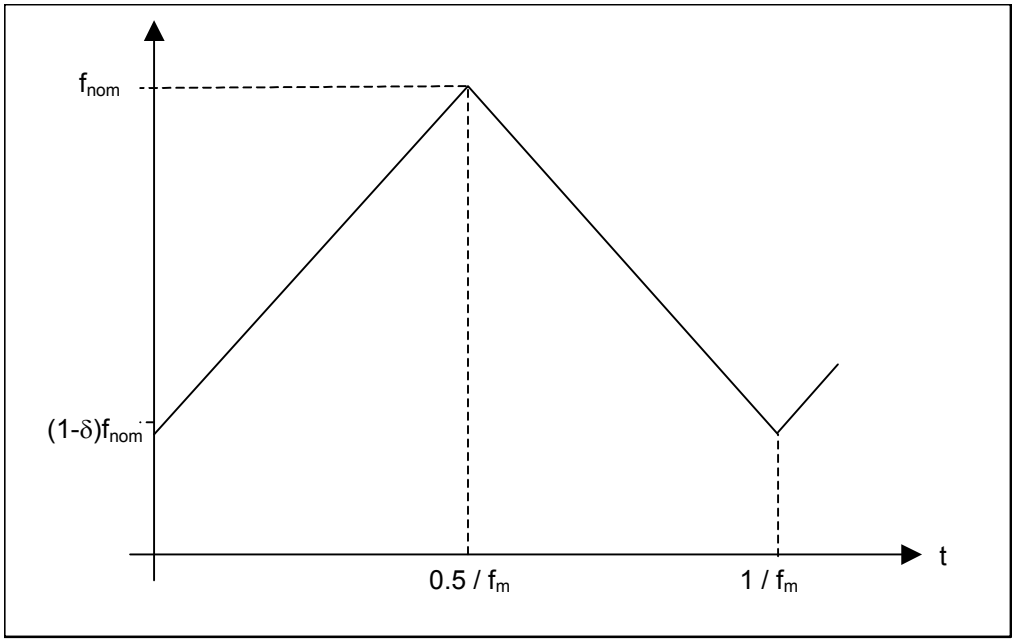


Figure 25 - Triangular frequency modulation profile

3. For triangular modulation, the clock frequency deviation (δ) is required to be no more than 0.5% “down-spread” from the corresponding nominal frequency, i.e., +0%/-0.5%. The absolute spread amount at the fundamental frequency is shown in Figure 26, as the width of its spectral distribution (between the –3 dB roll-off). The ratio of this width to the fundamental frequency cannot exceed 0.5%. This parameter can be measured in the frequency domain using a spectrum analyzer.

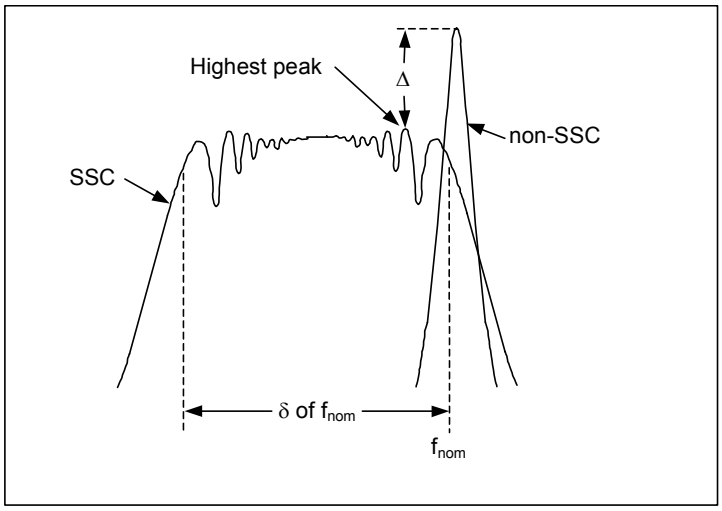


Figure 26 - Spectral fundamental frequency comparison

4. To achieved sufficient system-level EMI reduction, it is desired that SSC reduce the spectral peaks in the non-SSC mode by the amount specified in Table 14. The peak reduction Δ is defined, as shown in Figure 26, as the difference between the spectral peaks in SSC and non-SSC modes at the specified measurement frequency

Table 14 – Desired peak amplitude reduction by SSC.

Clock freq.	Peak reduction Δ	Measurement freq.
66 MHz	7 dB	466 MHz (7th harmonics)
75 MHz	7 dB	525 MHz (7th harmonics)

NOTES –
The spectral peak reduction is not necessarily the same as the system EMI reduction. However, this relative measurement gives the component-level indication of SSC's EMI reduction capability at the system level.

It is recommended that a spectrum analyzer be used for this measurement. The spectrum analyzer should have measurement capability out to 1 GHz. The measured SSC clock needs to be fed into the spectrum analyzer via a high-impedance probe compatible with the spectrum analyzer. The output clock should be loaded with 20 pF capacitance. The resolution bandwidth of the spectrum analyzer needs to be set at 120 KHz to comply with FCC EMI measurement requirements. The video band needs to be set at higher than 300 KHz for appropriate display. 100 KHz may be used as the resolution bandwidth in case of measurement equipment limitation. The display should be set with maximum hold. The corresponding harmonic peak readings should be recorded in both the non-SSC and the SSC modes, and be compared to determine the magnitude of the spectral peak reduction.

5. The modulation frequency of SSC is required to be in the range of 30-33 KHz to avoid audio band demodulation and to minimize system timing skew.

6.7 Functional specifications

6.7.1 Overview

The purpose of this section is to provide the normative functional definitions for various operations that a hardware platform is expected to accommodate.

6.7.2 Common-mode biasing

Since the Serial ATA protocol supports both direct coupled and AC-coupled solutions, there are four scenarios to be considered when applying the DC bias to TX and RX designs. In Figure 27 below, it is shown that only DC-coupled designs need sustain the specified 250 mV common-mode level to ensure interoperability.

A DC-coupled receiver (no blocking capacitors) shall weakly hold the common-mode level of its inputs at 250 mV. A DC-coupled transmitter shall transmit with the nominally specified 250 mV common-mode level.

An AC-coupled receiver shall meet the common mode AC coupled electrical requirements of Table 11and need not sustain the cable side of the its blocking capacitors. Similarly, an AC-coupled transmitter shall meet the common mode AC coupled electrical requirements of Table 11and need not sustain the common-mode level on the cable side of its blocking capacitors.

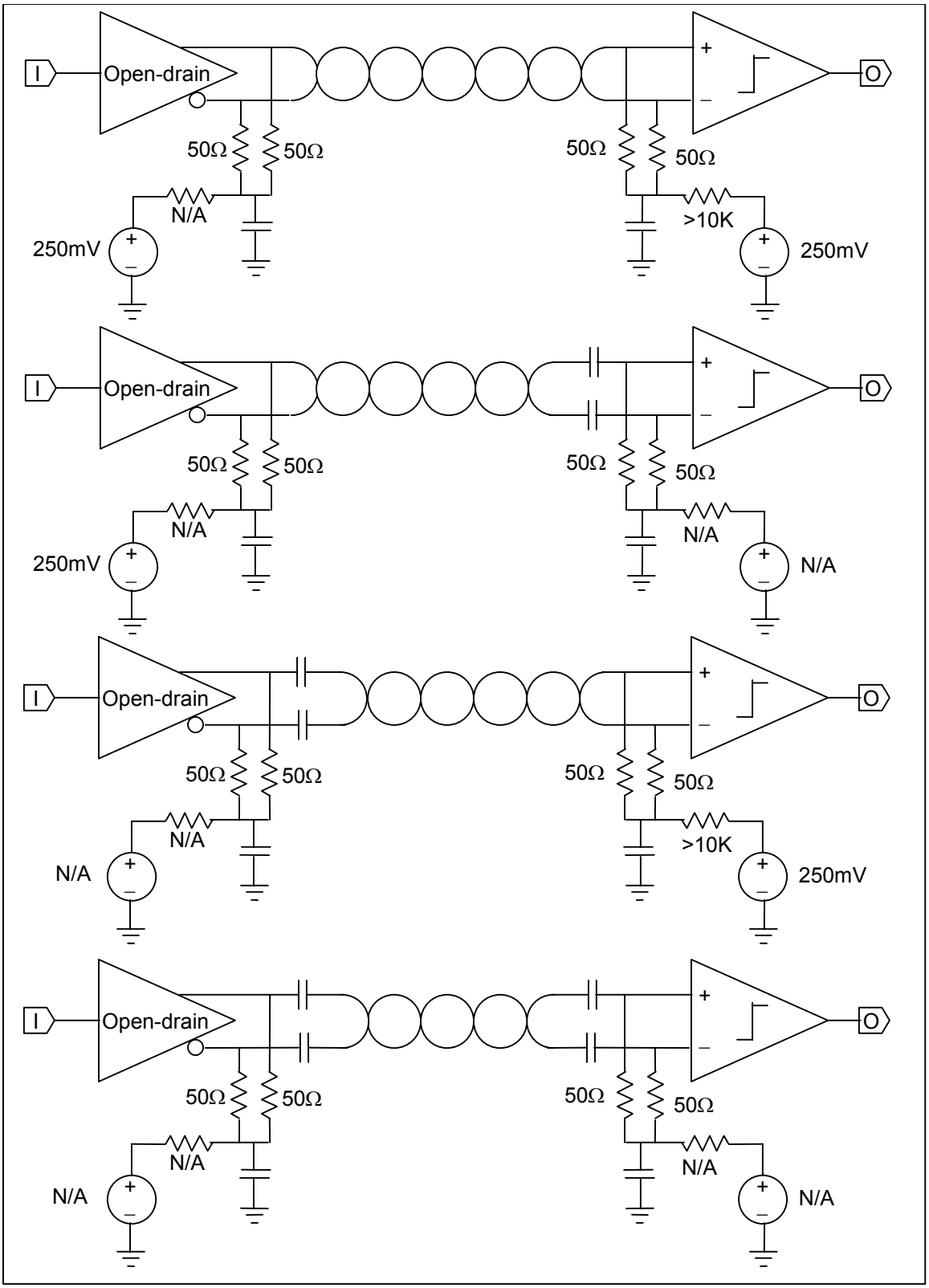


Figure 27 – Common-mode biasing

6.7.3 Matching

The host controller shall employ on-chip adaptive impedance matching circuits to ensure best possible termination for both its TX and RX. Device peripherals also shall provide impedance terminations, as per the specified parameters in Table 11 of section 6.6.2, and may adapt their termination impedance as well.

The host controller, since it is given the first opportunity to calibrate during the power on sequence, cannot assume that the far end of the cable is calibrated yet. For this reason, the host controller must utilize a separate reference to perform calibration. In a desktop system, the cable provides the optimal impedance reference for calibration.

Using Time Domain Reflectometry (TDR) techniques, the host may launch a step waveform from its transmitter, so as to get a measure of the impedance of the transmitter, with respect to the cable, and adjust its impedance settings as necessary.

In a mobile system environment, where the cable is small or non-existent, the host controller must make use of a separate reference (such as an accurate off-chip resistor) for the calibration phase.

The device, on the other hand, can assume that the termination on the far side (host side) of the cable is fully calibrated, and may make use of this as the reference. Using the host termination as the calibration reference allows the devices operating in both the desktop and the mobile system environment to use the same hardware.

Signals generated for the impedance calibration process shall not duplicate the OOB signals, COMWAKE, COMINIT, or COMRESET. Signals generated for the impedance calibration process shall not exceed the normal operating voltage levels, cited in section 6.6.2 See the power management section for suggested times to perform calibration during power-on.

6.7.4 Out of band signaling

There shall be three Out Of Band (OOB) signals used/detected by the Phy, COMRESET, COMINIT, and COMWAKE. COMINIT, COMRESET and COMWAKE OOB signaling shall be achieved by transmission of 160 Gen1 UI bursts of ALIGN primitives, each separated by idle periods (at common-mode levels), having durations of 160 Gen1 UI, as depicted in Figure 28, below. The content of the burst is reserved and set to be ALIGN primitives.

During OOB signaling transmissions, the differential and common mode levels of the signal lines shall comply with the same electrical specifications as for in-band data transmission, specified in section 6.6.2. In Figure 28 below, COMRESET, COMINIT, and COMWAKE are shown. OOB signals are observed by detecting the temporal spacing between adjacent bursts of activity, on the differential pair.

Any spacing less then 55 ns or greater then 175 ns shall invalidate the COMWAKE detector output. The COMWAKE OOB signaling is used to bring the Phy out of a power-down state (PARTIAL or SLUMBER) as described in section 6.8. The interface must be held inactive for at least 175ns after the last burst to ensure far-end detector detects the deassertion properly. The device may hold the interface inactive no more then 228.3ns (175ns + 2 Gen1 dwords) at the end of a COMWAKE to prevent susceptibility to crosstalk.

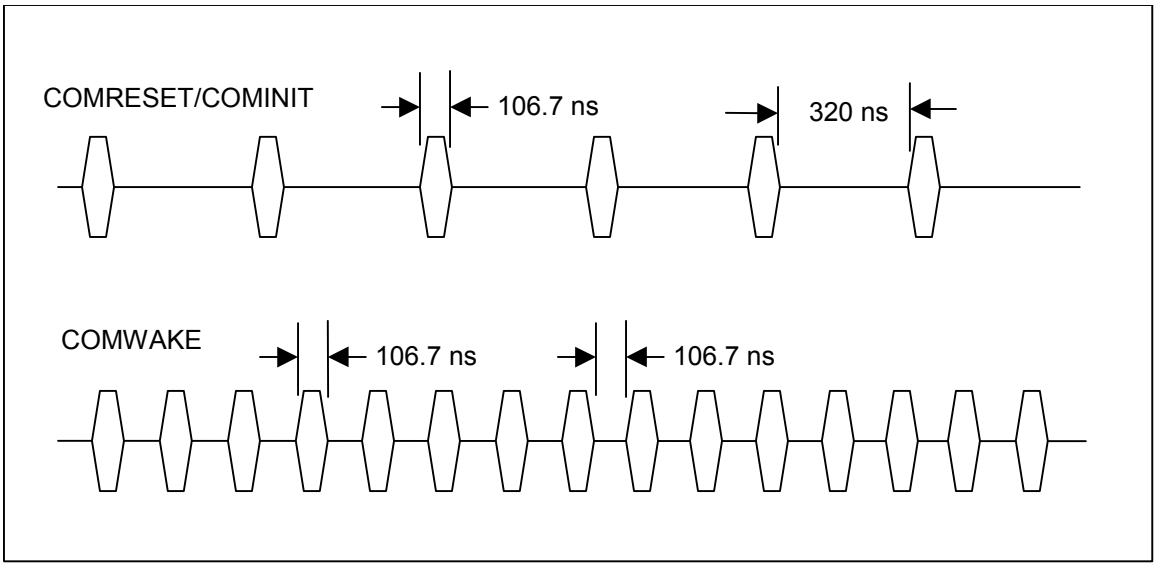


Figure 28 – Out of band signals

6.7.4.1 Idle bus status

During the idle bus condition, the differential signal diminishes to zero while the common mode level of 250 mV remains.

Common-mode transients, shall not exceed the maximum amplitude levels (Vcm,ac) cited in section 6.6.2, and shall settle to within 25 mV of Vcm,DC within Tsettle,CM, cited in in section 6.6.2. The following figure shows several transmitter examples, and how the transition to and from the idle state may be implemented.

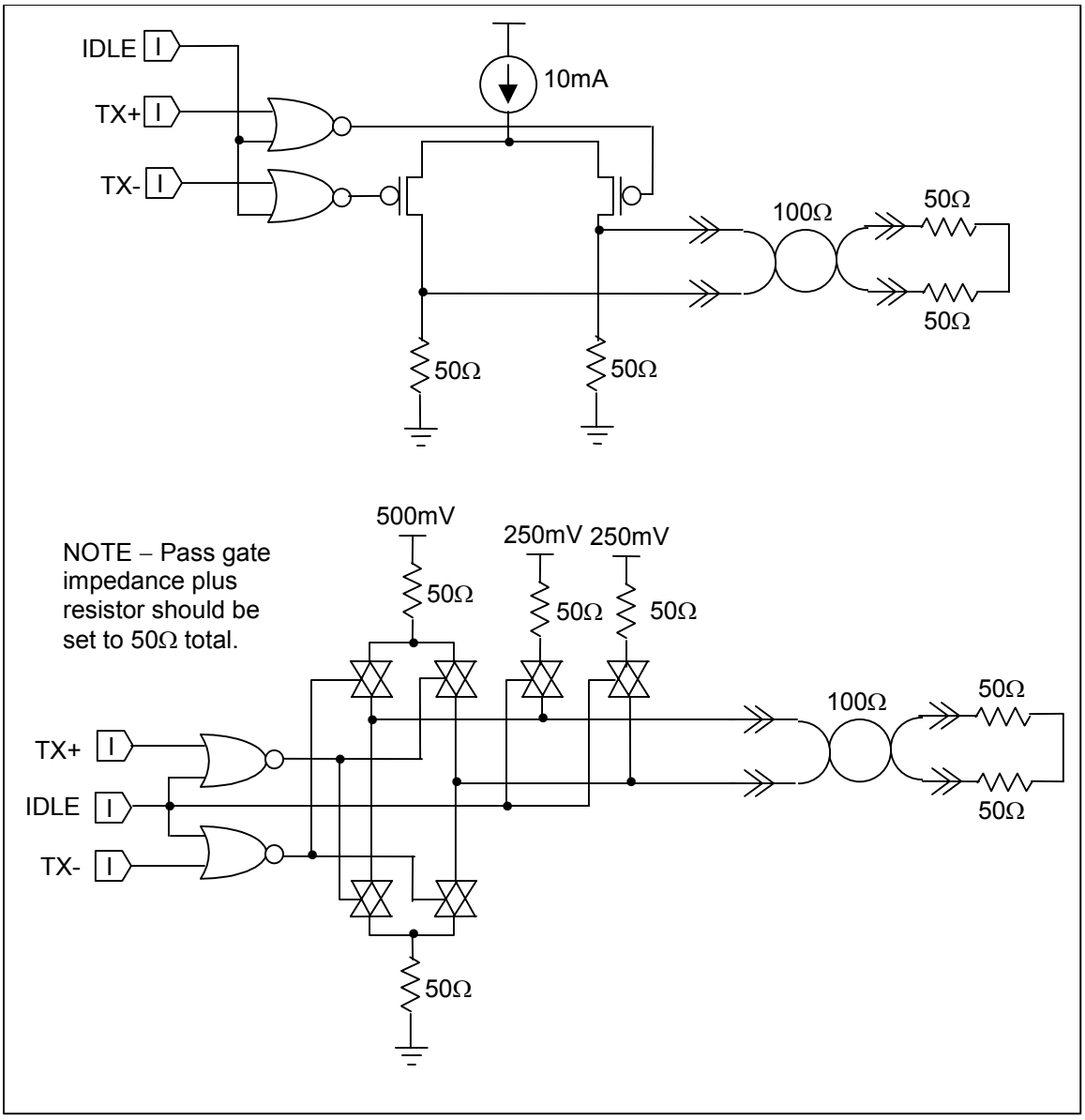


Figure 29 – Transmitter examples

(continued)

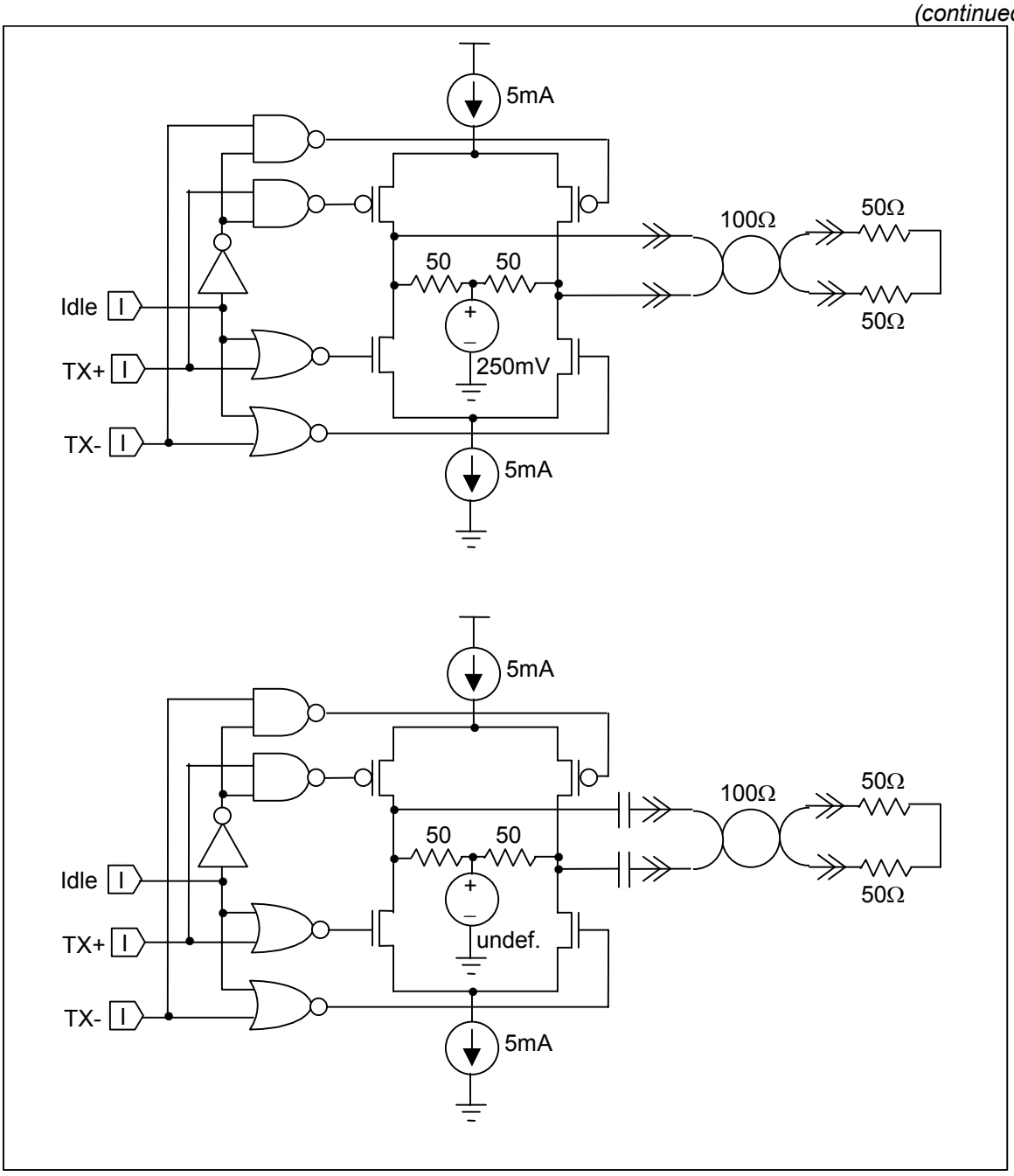


Figure 29 – Transmitter examples *(concluded)*

6.7.4.2 COMRESET

COMRESET always originates from the host controller, and forces a hard reset in the device. It is indicated by transmitting bursts of data separated by an idle bus condition.

The OOB COMRESET signal shall consist of no less than six data bursts, including inter-burst temporal spacing. The COMRESET signal shall be:

1) sustained/continued uninterrupted as long as the system hard reset is asserted, or
2) started during the system hard reset and ended some time after the deassertion of system hard reset, or
3) transmitted immediately following the deassertion of the system hard reset signal.

The host controller shall ignore any signal received from the device from the assertion of the hard reset signal until the COMRESET signal is transmitted.

Each burst shall be be 160 Gen1 UI's long (106.7 ns) and each inter-burst idle state shall be 480 Gen1 UI's long (320 ns). A COMRESET detector will look for four consecutive bursts with 320 ns spacing (nominal).

Any spacing less than 175 ns or greater than 525 ns shall invalidate the COMRESET detector output. The COMRESET interface signal to the Phy layer will initiate the Reset sequence shown in Figure 30 – COMRESET sequence below. The interface must be held inactive for at least 525ns after the last burst to ensure far-end detector detects the deassertion properly.

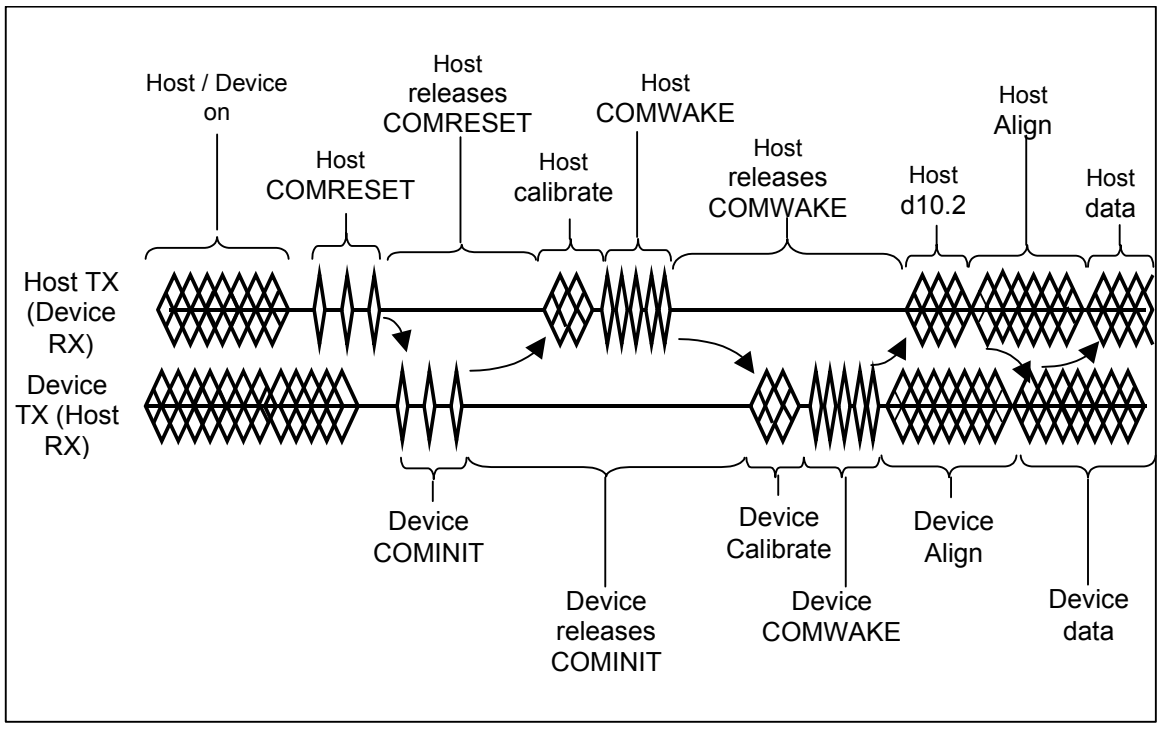


Figure 30 – COMRESET sequence

Description:

1. Host/device are powered and operating normally with some form of active communication.
2. Some condition in the host causes the host to issue COMRESET
3. Host releases COMRESET. Once the condition causing the COMRESET is released, the host releases the COMRESET signal and puts the bus in a quiescent condition.
4. Device issues COMINIT – When the device detects the release of COMRESET, it responds with a COMINIT. This is also the entry point if the device is late starting. The device may initiate communications at any time by issuing a COMINIT.
5. Host calibrates and issues a COMWAKE.
6. Device responds – The device detects the COMWAKE signal on its RX pair and calibrates its transmitter (optional). Following calibration, the device sends a six burst COMWAKE signal and then sends a continuous stream of the ALIGN sequence. After 2048 ALIGN Dwords have been sent without response from the host as determined by detection of ALIGN primitives received from the host, the device may assume that the host cannot communicate at that speed. If additional legacy speeds are available, the device will try the next fastest speed by sending 2048 ALIGN dwords at that rate. This step is repeated for as many legacy speeds are supported. Once the lowest speed has been reached without response from the host, the device will enter an error state.
7. Host locks – after detecting the COMWAKE, the host starts transmitting D10.2 characters (see 6.7.6) at its lowest supported rate. Meanwhile, the host receiver locks to the ALIGN sequence and, when ready, returns the ALIGN sequence to the device at the same speed as received. A host must be designed such that it can acquire lock given 2048 ALIGN Dwords. The host should allow for at least 32768 Gen1 dwords (880us) after detecting the release of COMWAKE to receive the first ALIGN. This will ensure interoperability with multi-generational and synchronous devices. If no ALIGN is received within 32768 Gen1 dwords (880us), the host shall restart the power-on sequence – repeating indefinitely until told to stop by the application layer.
8. Device locks – the device locks to the ALIGN sequence and, when ready, sends the SYNC primitive indicating it is ready to start normal operation.
9. Upon receipt of three back-to-back non-ALIGN primitives, the communication link is established and normal operation may begin.

6.7.4.3 COMINIT

COMINIT always originates from the drive and requests a communication initialization. It is electrically identical to the COMRESET signal except that it originates from the device and is sent to the host. It is used by the device to request a reset from the host in accordance to the sequence shown in Figure 31, below.

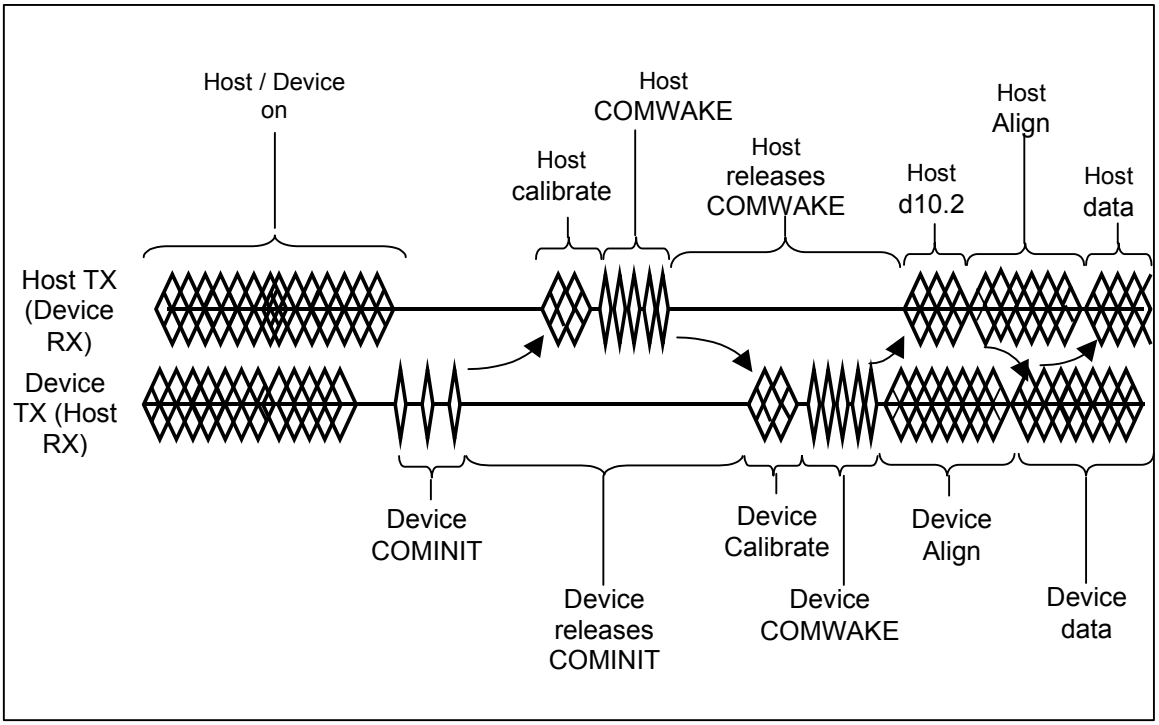


Figure 31 – COMINIT sequence

Description:

1. Host/device are powered and operating normally with some form of active communication.
2. Some condition in the device causes the device to issues a COMINIT
3. Host calibrates and issues a COMWAKE.
4. Device responds – The device detects the COMWAKE signal on its RX pair and calibrates its transmitter (optional). Following calibration, the device sends a six burst COMWAKE signal and then sends a continuous stream of the ALIGN sequence. After 2048 ALIGN Dwords have been sent without response from the host as detected by the envelope detector, the device may assume that the host cannot communicate at that speed. If additional legacy speeds are available, the device will try the next fastest speed by sending 2048 ALIGN dwords at that rate. This step is repeated for as many legacy speeds are supported. Once the lowest speed has been reached without response from the host, the device will enter an error state.
5. Host locks – after detecting the COMWAKE, the host starts transmitting D10.2 characters (see 6.7.6) at its lowest supported rate. Meanwhile, the host receiver locks to the ALIGN sequence and, when ready, returns the ALIGN sequence to the device at the same speed as received. A host must be designed such that it can acquire lock given 2048 ALIGN Dwords. The host should allow for at least 32768 Gen1 dwords (880us) after detecting the release of COMWAKE to receive the first ALIGN. This will ensure interoperability with multi-generational and synchronous devices. If no ALIGN is received within 32768 Gen1 dwords (880us), the host shall restart the power-on sequence – repeating indefinitely until told to stop by the application layer.
6. Device locks – the device locks to the ALIGN sequence and, when ready, sends the SYNC primitive indicating it is ready to start normal operation.
7. Upon receipt of three back-to-back non-ALIGN primitives, the communication link is established and normal operation may begin.

6.7.4.4 COMWAKE

COMWAKE can originate from either the host controller or the device. It is signaled by transmitting six bursts of data separated by an idle bus condition.

The OOB COMWAKE signaling shall consist of no less than six data bursts, including inter-burst temporal spacing.

Each burst shall be 160 Gen1 UI's long and each inter-burst idle state shall be 160 Gen1 UI's long. A COMWAKE detector will look for four consecutive burst with a 106.7 ns spacing (nominal).

Any spacing less than 55 ns or greater than 175 ns shall invalidate the COMWAKE detector output. The COMWAKE OOB signaling is used to bring the Phy out of a power-down state (PARTIAL or SLUMBER) as described in section 6.8. The interface must be held inactive for at least 175ns after the last burst to ensure far-end detector detects the deassertion properly. The device may hold the interface inactive no more then 228.3ns (175ns + 2 Gen1 dwords) at the end of a COMWAKE to prevent susceptibility to crosstalk.

6.7.4.5 Design example

This section is informative and represents one possible design example for detecting COMRESET/COMINIT and COMWAKE. Other design implementations are possible as long as they adhere to the requirements listed in this specification.

The output of the squelch detector is fed into four frequency comparators. When the period is within the window determined by the RC time constants for three consecutive cycles, the appropriate signal is asserted.

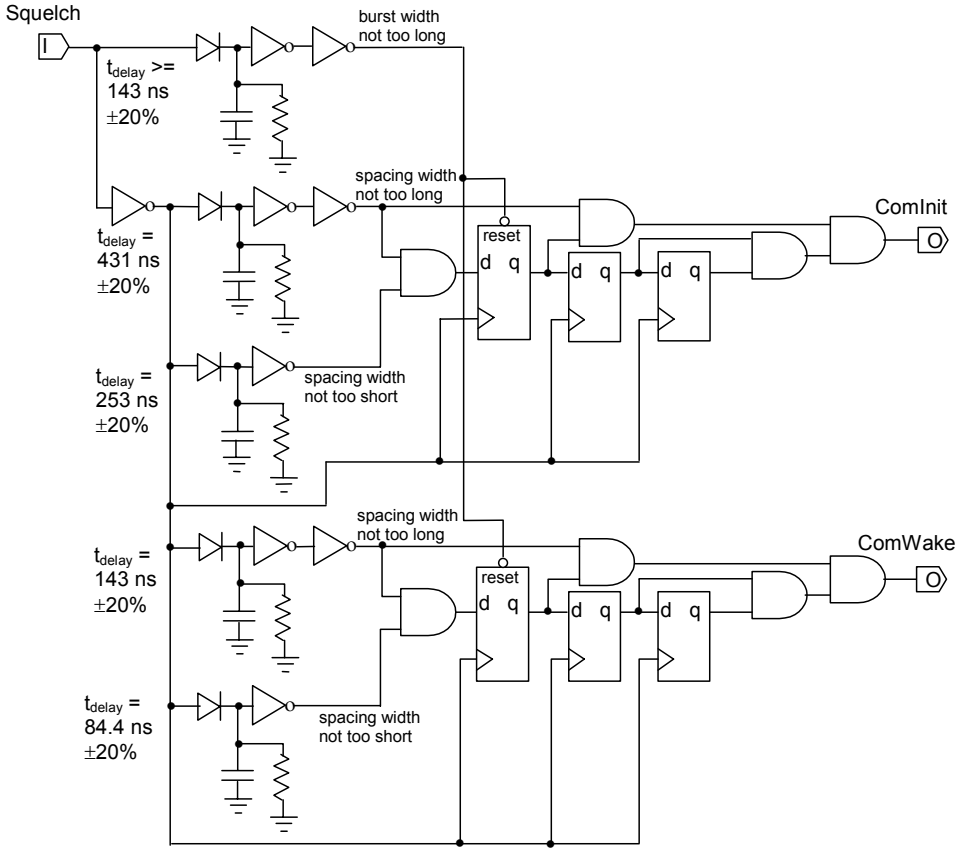


Figure 32 – OOB signal detector

The Squelch detector example below makes use of a receiver with built-in hysteresis to filter out any signal not meeting the minimum amplitude. The squelch detector receiver must be true differential to ensure common-mode noise is rejected.

The full-swing output is fed into a pulse generator that charges up the capacitor through the diode. In the absence of signal, a resistor discharges the capacitor to ground. The circuit outputs a true signal when the capacitor voltage is below the turn-on threshold of the Schmitt trigger buffer – indicating insufficient signal level. This circuit must be enabled in all power management states and should, therefore, be implemented with a small power budget.

Figure 33, like the OOB Signal Detector figure shown in Figure 32, is intended to show functionality (informative) only, and other solutions may be used to improve power consumption as long as they comply to the electrical specifications of section 6.6.2, for the worst case noise environment (common-mode) conditions.

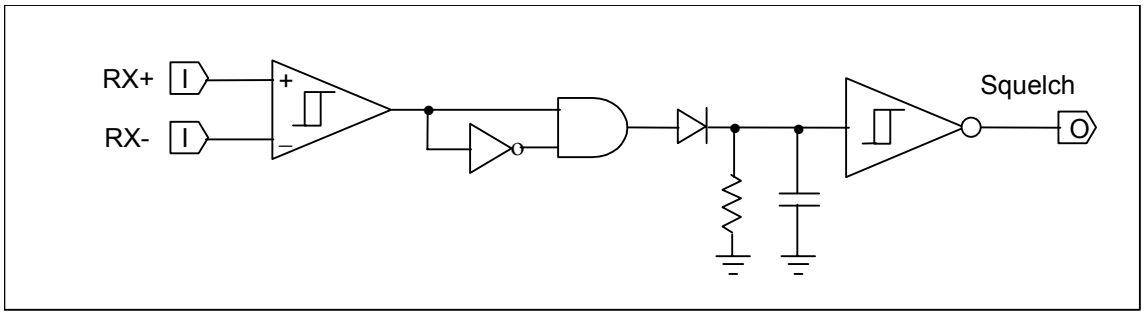


Figure 33 – Squelch detector

6.7.5 Idle bus condition

During power management states (Partial and Slumber), the electrical interface shall maintain the proper common-mode levels, as cited in section 6.6.2, with zero differential on both signal pairs (all four conductors at 250 mV) for all interface scenarios, except for the case where both, the device and the host-controller, are AC-coupled and the conductor pairs are allowed to float.

All transmitter designs must ensure that transition to and from the idle bus condition do not result in a disturbance in the differential baseline on the conductors. To accomplish this, an AC-coupled transmitter must hold its outputs at zero differential with the same common-mode level as normal operation when in the partial power management mode. When operating in the slumber power management mode, the common mode level of the AC coupled transmitter is allowed to float (while maintaining zero differential) as long as it remains within the limits cited in section 6.6.2.

It is unacceptable to hold the TX outputs at a logical zero or one state during the idle bus condition since this will result in a baseline shift when communications are resumed.

6.7.6 Elasticity buffer management

For non-tracking implementations elasticity buffer circuitry may be required to absorb the slight differences in frequencies between the host and device. The greatest frequency difference result from a SSC compliant device talking to a non SSC device. The average frequency difference will be just over 0.25% with excursions as much as 0.5%.

The Serial ATA standard is written to support both tracking and non-tracking architectures. A non-tracking architecture shall contain the elasticity buffer within the Phy layer.
Note that since this elasticity buffer will be designed to have finite length, there needs to be a mechanism at the physical layer protocol level that allows this receiver buffer to be reset without dropping or adding any bits to the data stream. This is especially important during reception of long continuous streams of data. This physical layer protocol must not only support oversampling architectures but must also accommodate unlimited frame sizes (the frame size is limited by the CRC polynomial).

The Link Layer shall keep track of a resettable counter that rolls over at most every 1024 transmitted characters (256 Dwords). Prior to, or at the pre-roll-over point (all 1's), the Link Layer shall trigger the issuance of dual, consecutive ALIGN primitives which shall be included in the Dword count.

After communications have been established, the first and second words out of the Link Layer shall be the dual-ALIGN primitive sequence, followed by at most 254 non-ALIGN Dwords. The cycle repeats starting with another dual-consecutive ALIGN primitive sequence. The Link may issue more

than one dual ALIGN primitive sequence but shall not send an unpaired ALIGN primitive (i.e. ALIGN primitives are always sent in pairs) except as noted for retimed loopback.

The ALIGN primitive consists of the following four characters

(rd+)	(rd-)	
1100000101	0011111010	Align1 (K28.5)
0101010101	0101010101	Align2 (D10.2)
0101010101	0101010101	Align3 (D10.2)
1101100011	0010011100	Align4 (D27.3)

6.7.7 Test considerations

6.7.7.1 Physical plant as a system

A number of components of the system make up the system's performance, not just by mutually exclusively summing up the low-level error parameters detected at the bottom of the protocol stack can be made, but also including the retry algorithms.

There are two basic classes of errors that will affect the bit-error rate performance that have to be considered: bit-errors, and burst-errors. The following section addresses the methodology, for in-system bit-error-rate computations.

For in-system testing, the criteria for testing to a specified maximum frame error rate will be addressed, using a set of reference frames, defined by a specific set of ordered test patterns within the frame.

6.7.7.1.1 Test bit patterns and sequence characteristics

Test bit sequences are those bit sequences that are transmitted onto a serial link, so as to test the Serial ATA interface's jitter compliance, as well as its derived communications-link performance.

As discussed earlier, jitter is generally classified as Random Jitter (RJ), and Deterministic Jitter (DJ). RJ is Gaussian in nature, adds on a power basis, and is measured as an RMS value. DJ is also referred to as systematic jitter, and usually is introduced into the serial link due to duty cycle distortion, power supply noise, substrate noise, or inter-symbol interference.

We will focus on various types of bit sequence patterns that emphasize low/high transition density patterns, as well as low/high frequency patterns.

a) Low transition density patterns are those patterns containing long runs of ones and zeroes, intended to create inter-symbol interference by varying the excursion times at either extreme of the differential signaling levels.

b) High transition density patterns are those patterns containing short runs of ones and zeroes, also intended to create inter-symbol interference.

c) Bit patterns that contain low frequency spectral components are a good test of the input high pass filter circuitry, more specifically, introduced amplitude signal distortion, due to a marginal design. These bit patterns are a better test than those bit patterns having high frequency spectral content.

d) Simultaneous switching outputs patterns are achieved by transmitting alternating 1's complement bit patterns (10-bits) for recovery at the receiver. These patterns create worst case power supply, or chip substrate, noise, and are achieved by selecting bit test pattern sequences that maximize current extremes at the recovered bit pattern parallel interface. These patterns induce Ldi/dt noise into substrate supply, and are a good test of the receiver circuitry.

e) The intent of random bit patterns is to provide those patterns containing sufficiently broad spectral content, and minimal peaking, that can be used for both component, and system level architecture measurement of jitter output, and bit-error-rate performance. These patterns are also intended to be the common baseline pattern stimulus, for system/component vendor comparative testing, attributing the Transmit jitter output measurement to the component performance, and not to the spectral profile of the data pattern used.

These patterns will be used for compliance testing of the Serial ATA interfaces. The patterns will be classified in two categories:

a) Non-compliant patterns
b) Compliant patterns

Non-compliant patterns are those patterns that are used for baseline jitter measurements, and assessment of signal quality, given specified stimulus. These patterns do not comply to the required FIS formats, but are just a repeated selected set of 8b/10b characters.

Compliant patterns are those specified patterns that do contain the leading SOF primitive, the specified pattern as data content, and trailing CRC, and EOF primitives. There is no suppression of the dual-consecutive ALIGN primitive during stimulus with this class of pattern.

The test patterns illustrated in the following sections are indicated to start with negative running disparity for illustrative purposes only in order to convey the encoded 10b patterns transmitted on the wire for each sequence.

6.7.7.1.1.1 Low transition density bit pattern sequences

Low transition density bit patterns, as shown in Figure 34 below, are those patterns containing long runs of ones and zeroes. These patterns create high frequency jitter due to inter-symbol interference, emphasized further when part of the composite pattern described in a later section.

Low transition density pattern (SATA)

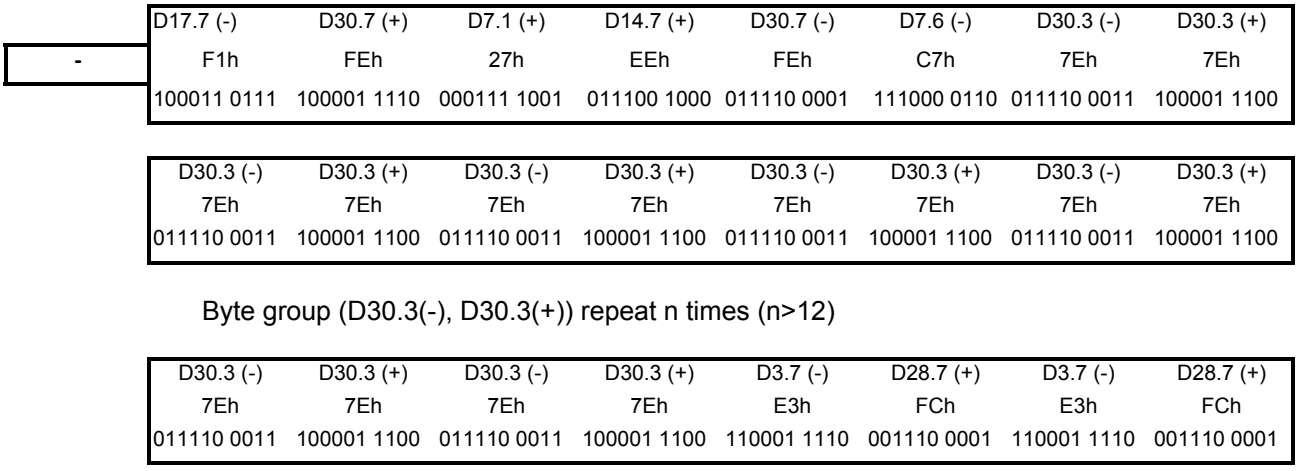


Figure 34 – Low transition density pattern

6.7.7.1.1.2 High transition density bit pattern sequences

High transition density patterns are those patterns containing short runs of ones and zeroes, also intended to create high frequency jitter due to inter-symbol interference, emphasized further when part of the composite pattern described in a later section. For this case there are two sub-classes of high-transition density patterns:

a) Half-rate high transition density bit pattern sequence
b) Quarter-rate high transition density bit pattern sequence

An amalgamate of the two sub-classes of high-transition density bit patterns will be used to represent high transition density test pattern.

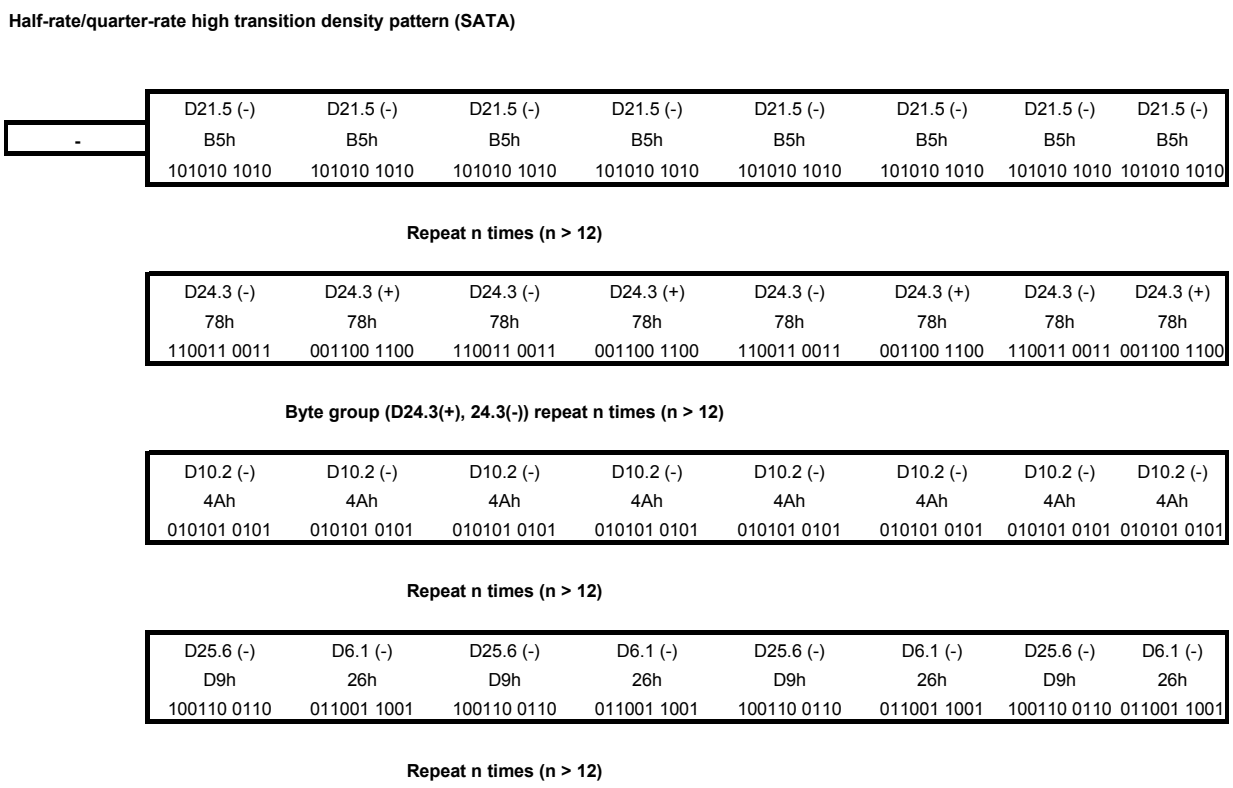


Figure 35 - Half-rate / quarter-rate high transition density pattern

6.7.7.1.1.3 Low frequency spectral content bit pattern sequences

Bit patterns that contain low frequency spectral components are a good test of the input high pass filter circuitry, as far as introduced signal distortion, due to a marginal design.

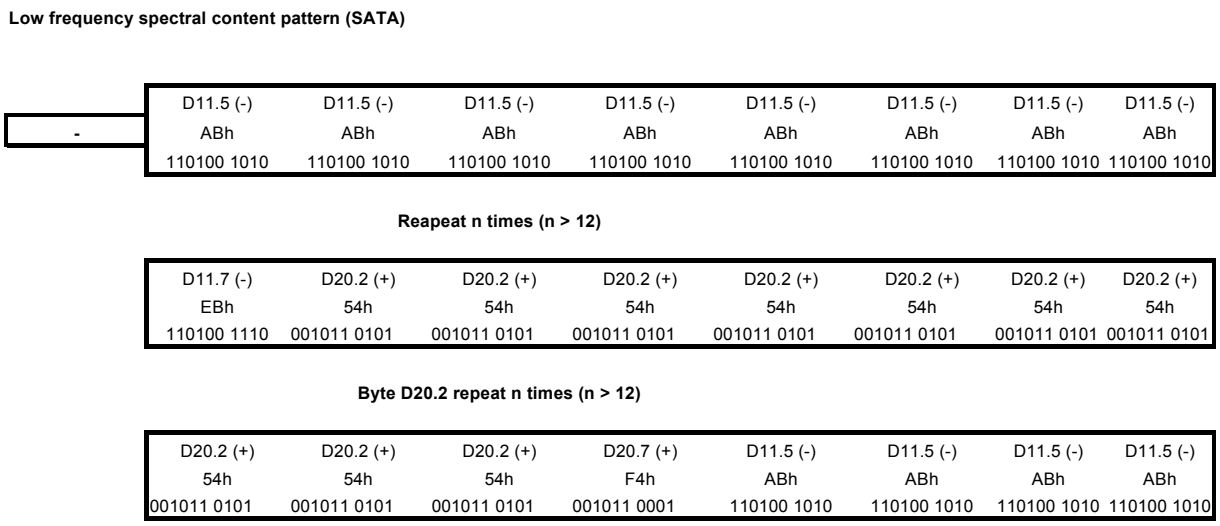


Figure 36 – Low frequency spectral content pattern

6.7.7.1.1.4 Simultaneous switching outputs bit pattern sequences

Simultaneous switching outputs bit pattern sequences induce Ldi/dt noise into substrate supply, and is a good test of the receiver circuitry. This is achieved by transmitting alternating 1's complement bit patterns (10-bits) for recovery at the receiver.

Simultaneous switching outputs patterns (SATA)

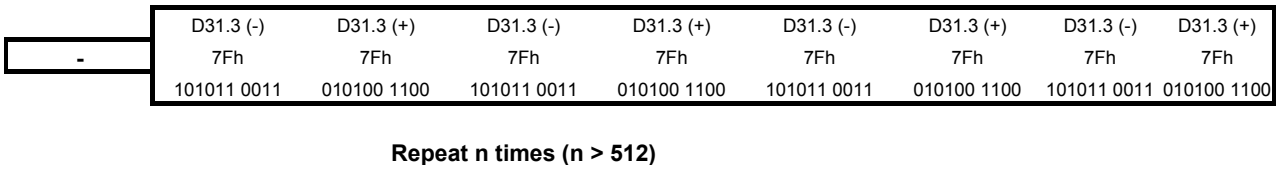


Figure 37 – Simultaneous switching outputs patterns

6.7.7.1.1.5 Composite bit pattern sequences

For the measurement of jitter, patterns should combine low frequency, low transition density, and high transition density patterns. All these combinations, but the Low Frequency Spectral Content class can be performed for relatively short test time intervals, for jitter and performance measurements.

The lower frequency pattern needs to be tested for longer interval periods to be able to observe the lower frequency jitter effects on the interface.

By including a composite set of the cited patterns, the resultant composite pattern stresses the interface components within the link with low and high frequency jitter, tests for component, and various amplitude distortions due to marginal receiver input circuitry, or interface components.

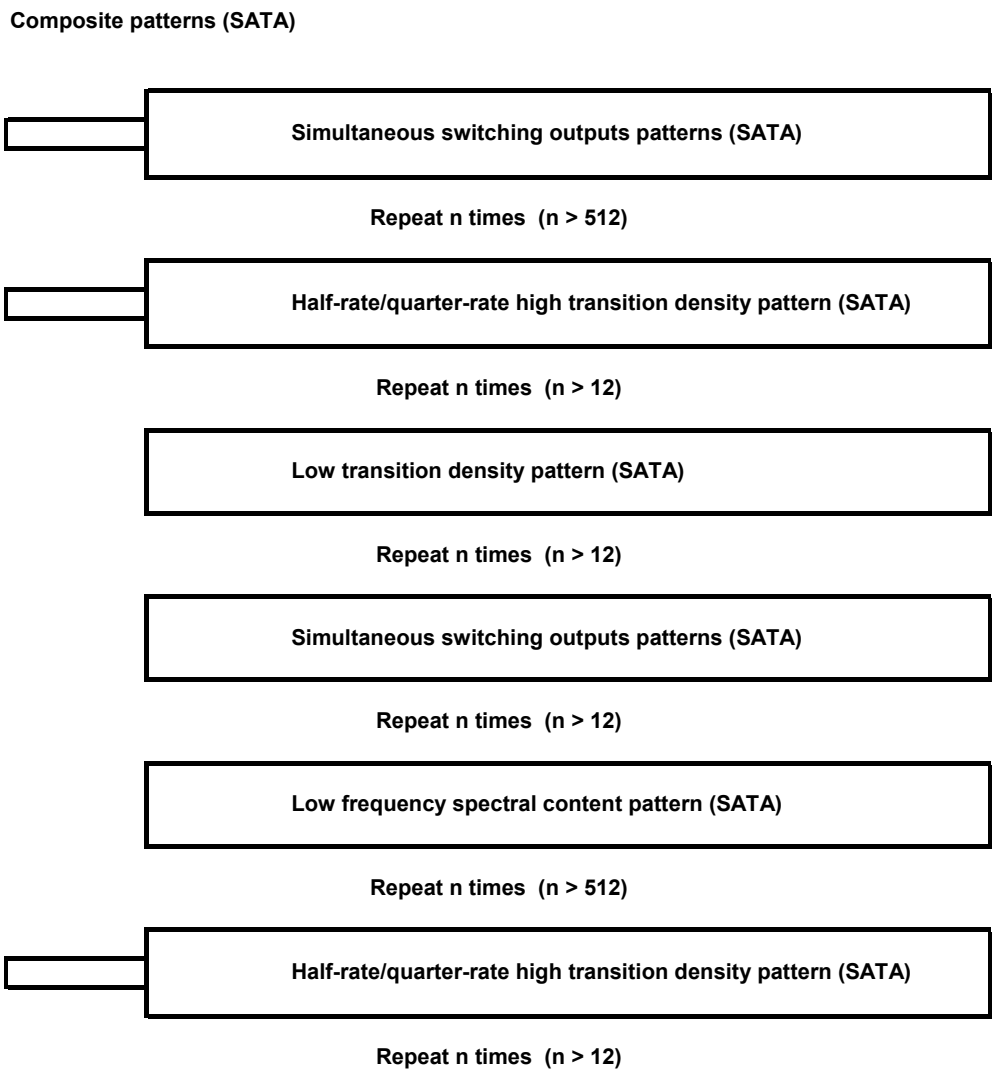


Figure 38 – Composite patterns

6.7.7.1.2 Bit error rate testing - Informative

In order to get a fair assessment of bit-error-rate performance, bit-errors, as well as burst errors, must be considered separately.

Note that a single bit error has a high probability of causing a byte-wise error, or an 8b/10b code violation error, due to the 8b/10b encoding, thus a single bit error translates to 8-bits or 10-bits of error. We will classify each of these occurrences, as "byte errors", and these form the basis for the bit-error-rate calculations.

Furthermore, a missing or an extra bit detected by the receiver will, translate into a series of errors that spans across multiple byte-boundaries until bit re-alignment via ALIGN primitives as per 6.7.6 - Elasticity buffer management, worst case. This series of errors will be defined as burst errors.

A third type of byte-wise error exists where, an entire byte is not received, this type of error will not be classified differently, as it will cause a burst error whose span can be limited by higher-layer protocol transmission conventions, at the cost of additional overhead ALIGN primitive sequences.

A single error may result in several related errors occurring closely together which in turn may result in multiple bit-error counts. A character might have a single bit error in it that causes a code-violation error. A disparity error might occur on a following character, caused by the same single error. To prevent multiple error counts from a single cause, the following concept of an error burst is introduced:

- An “error burst” is defined as a time period 1.5 +/- 0.5 seconds long, during which one or more invalid transmission words are recognized. This time may be exceeded due to infrequent unusual conditions.

- Only one error in an error burst shall be counted toward the error-burst-rate threshold.

6.7.7.1.2.1 Error-burst-rate-thresholding measurement - Informative

The error-burst-rate thresholding process is designed to detect an increased error rate before performance degradation becomes serious. Measurement of error-burst-rate thresholding is accomplished by counting the number of error bursts that occur in a 5-minute period, when tested with the compliant frame patterns, defined by section 6.7.7.1.1.1, but with the N parameters extended so as to achieve the maximum frame length. When the count equals a specific number, the threshold is exceeded. This specific number is called the “threshold error count.”

An error-burst-rate threshold is detected when 15 error bursts occur in a 5-minute period. The required accuracy of the 5-minute period mentioned above is +/-0.3 seconds. Reaching the threshold error count of 15 in a 5-minute period is a positive indication of the interface failing the error rate requirements and may be used to terminate testing. However, not reaching the threshold may not be taken as indication that the interface is passing without also confirming that the frame error rate requirements are satisfied using statistically sound measurement techniques. Being under the threshold error count after a 5-minute period should therefore never be used as a measure of system compliance.

The error-burst-rate counting process will be restarted when PHY Ready state is entered, and when an amount of time has elapsed after a error-burst-rate threshold is detected. After an error-burst-rate threshold is detected, at least 15 additional error bursts will occur before the next error-burst-rate threshold is detected. In addition, the error-burst counting process may be restarted whenever the 5-minute time period has expired even though an error-burst-rate threshold is not reached.

6.7.7.1.2.2 Bit-error-rate measurements - Informative

Bit error rates if measured and computed, byte-wise, should be no greater than 10^{-12} bit-errors when tested with the reference test patterns, cited in section 6.7.7.1.1.

Note that the Frame-Error Rate measurements constitute the basis for the applicable test requirements for this specification.

6.7.7.1.3 Frame error rate testing

Frame error rate testing is the measure of link performance using all the intermediate circuit blocks in the chain from low-level Phy Layer, Link Layer, through Transport Layer. Error detection is at the frame level using the CRC (Cyclic Redundancy Check) error detection mechanism, and respective reporting to the higher layer levels..

6.7.7.1.3.1 Frame error-rate patterns

Frame error rate patterns will contain the elements of the test bit patterns and sequence of patterns, so as to thoroughly be able to test the serial interface in system, using the higher level CRC error detection/reporting from the lower protocol level layers to the Application level layer.

The frame patterns shall be comprised of the set of the Composite Patterns, cited in section 6.7.7.1.1.5, but with the N parameters extended so as to achieve the maximum frame length.

6.7.7.1.3.2 Frame error-rate measurements

Frame Error Rates shall be measured and computed, framewise, to be no greater than ($8.196*10^{-8}$) frame errors when tested with any given 8b/10b pattern, including the Frame-Error-Rate reference patterns cited in section 6.7.7.1.3.1.

Note that the cited patterns should appear on the wire, and the N parameters of the reference patterns shall be extended to achieve the maximum frame length, as specified in paragraph 7.5.1.

6.7.7.1.4 Loopback testing

Three types of Loopback test schemes have been identified, where certain of these will not form part of this specification, but will be classified as Vendor Specific.

a) Far-End Retimed - Required
b) Far-End Analog - Vendor Specific
c) Near-End Analog (Effectively Retimed) - Vendor Specific

6.7.7.1.4.1 Loopback -- Far end retimed

Figure 24 below, illustrates the scope, at the architectural block diagram level, of the Far-End Retimed loopback. As this loopback scheme needs a specific action from the far-end connected interface, this mode shall be entered by way of the BIST FIS described in section 8.5.7.

The Far-End Interface shall remain in this Far-End Retimed Loopback, until receipt of the COMRESET/COMINIT OOB Signaling sequence.

As a minimum, Far-End Retimed Loopback shall involve far-end circuitry such that the datastream, at the Far-End interface, is extracted by the Serializer/Deserializer (SerDes) and data recovery circuit (DRC) before being sent back through the SerDes and Transmitter with appropriately inserted retiming ALIGN primitives as described in section 8.5.6.1. The data may be decoded and descrambled in order to provide testing coverage for those portions of the device, provided the data is re-scrambled using the same sequence of scrambler syndromes. The returned data shall be the same as the received data with the exception that the returned data may be encoded with different starting running disparity.

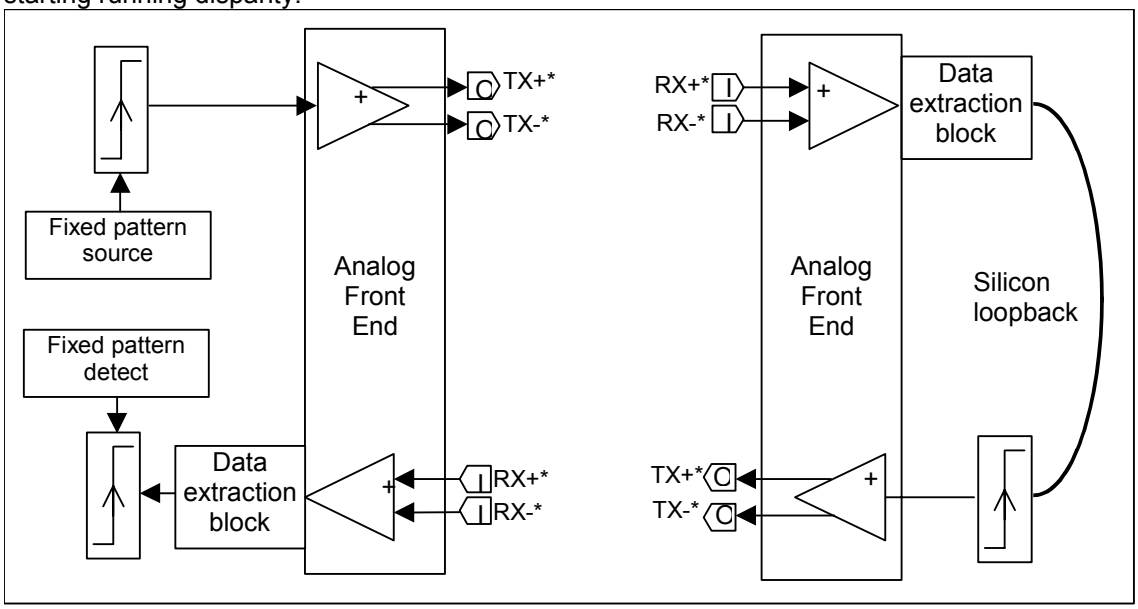


Figure 39 – Loopback far-end retimed

6.7.7.1.4.2 Loopback -- far-end analog (vendor specific)

Figure 40 below, illustrates the scope, at the architectural block diagram level, of the Far-End Analog loopback. As this loopback scheme needs a specific action from the far-end connected interface, this mode, if implemented, shall be entered by way of the BIST FIS described in section 8.5.7.

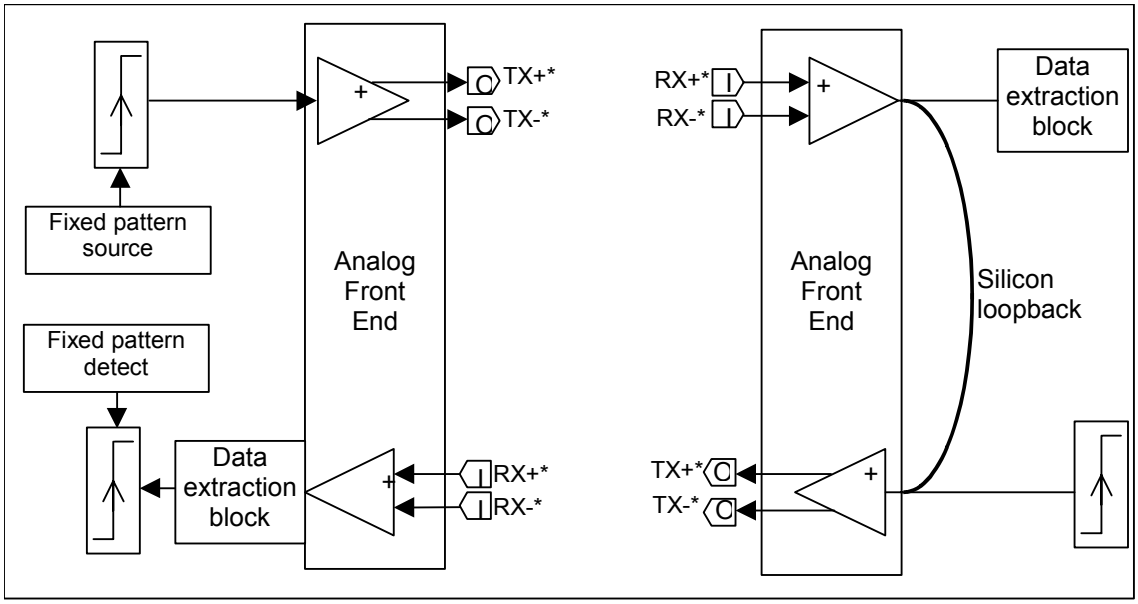


Figure 40 – Loopback far-end analog

If implemented, the Far-End Interface shall remain in this Far-End Analog Loopback mode, until receipt of the COMRESET/COMINIT OOB Signaling sequence.

6.7.7.1.4.3 Loopback -- near-end analog (vendor specific)

Figure 41 below, illustrates the scope, at the architectural block diagram level, of the Near-End Analog loopback. This loopback scheme, if implemented, needs the far-end connected interface to be in a non-transmitting mode, such as Slumber, or Partial interface power management states.

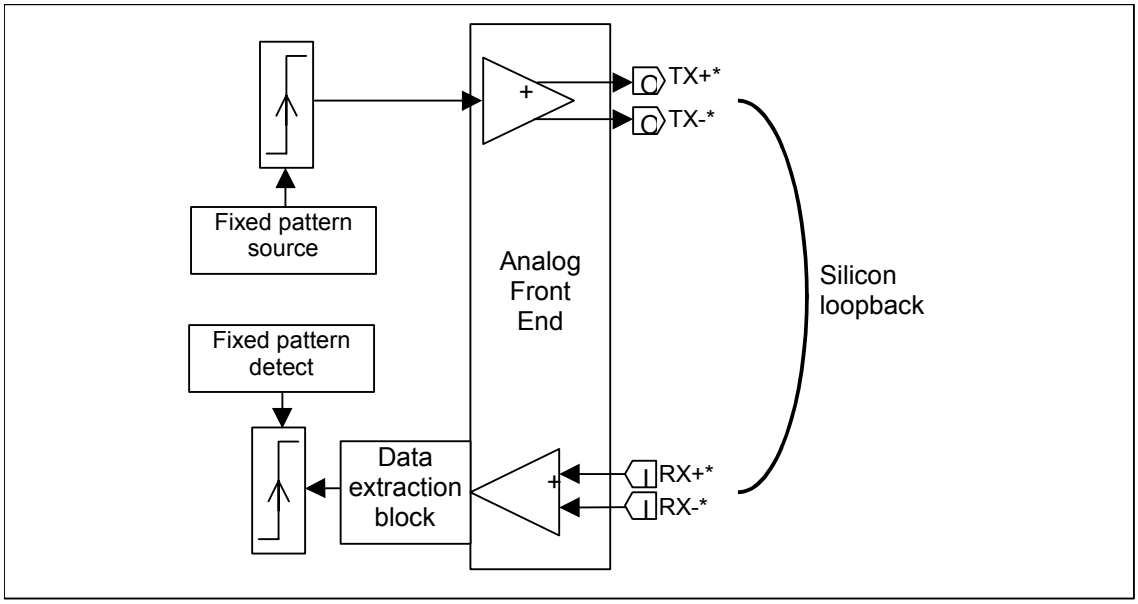


Figure 41 – Loopback - near-end analog

6.7.7.1.5 Test requirements

Test patterns cited in this section used as stimulus that will be used to verify the Serial ATA interface compliance and signal integrity, using the following test models:

a) Non-compliant test patterns for jitter measurements, physical connection media tests, and electrical parameter testing.
b) Compliant test patterns for frame error rate testing and in-system tests.

6.7.7.1.5.1 Test requirements - non-compliant patterns

Electrical parameters of section 6.6.2 shall be verified using the following set of test patterns, using the BIST FIS, Far-End Transmit Mode, described in section 8.5.7:

a) Low transition density bit patterns, as per section 6.7.7.1.1.1
b) High transition density bit patterns as per section 6.7.7.1.1.2
c) Low frequency spectral component bit patterns as per section 6.7.7.1.1.3
d) Simultaneous switching outputs bit patterns as per section 6.7.7.1.1.4

6.7.7.1.5.2 Test requirements - compliant frame patterns

The frame error rates specified in section 6.7.7.1.3.2 shall be tested for compliance when subjected to any implementation-determined worst-case compliant patterns, as well as the following set of compliant patterns:

a) Compliant low transition density bit patterns, as per section 6.7.7.1.1.1
b) Compliant high transition density bit patterns as per section 6.7.7.1.1.2
c) Compliant low frequency spectral component bit patterns as per section 6.7.7.1.1.3
d) Compliant simultaneous switching outputs bit patterns as per section 6.7.7.1.1.4
e) Compliant composite patterns as per section 6.7.7.1.1.5

Where the qualifying prefix term "compliant" signifies transmission of the cited pattern encapsulated in the data portion of the Frame Information Structure, and used in a Serial ATA operational transmission context.

Note that the cited patterns should appear on the wire, and the N parameters of the reference patterns shall be extended to achieve the maximum frame length. These compliant patterns contain the necessary SOF leading primitive, the calculated CRC, and the trailing EOF primitive, as shown in the Figure below:

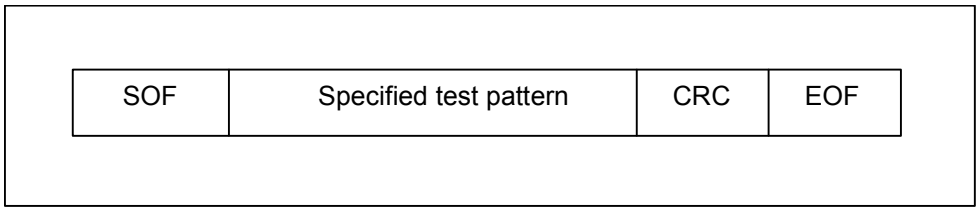


Figure 42 – Compliant test patterns

6.7.7.1.6 Test requirements - loopback

As specified earlier in section 6.7.7.1.4, only one of the three types of Loopback test schemes will form part of this specification: Far-End Retimed Loopback. Both Far-end and Near-End Analog Loopback schemes are classified as Vendor Specific, thus will not have specified test requirements, herein.

6.7.7.1.6.1 Test requirements - loopback - far-end retimed

Generation of loopback test patterns is optional and vendor specific. As this loopback scheme needs a specific action from the far-end connected interface, this mode shall be entered by way of the BIST FIS described in section 8.5.7.

This loopback test is intended for a relatively quick assessment of interface health, accommodating an in-system test capability,

In this Far-End Retimed Loopback mode, the interface shall operate without a CRC error for a sustained minimum period of 1000 ms at Gen1 speed, using the following set of compliant reference frame patterns if the test pattern generation is supported by the initiating device:

a) Compliant composite reference frame patterns as specified in section 6.7.7.1.3.1.
b) Compliant low transition density bit patterns, as per section 6.7.7.1.1.1
c) Compliant high transition density bit patterns as per section 6.7.7.1.1.2.
d) Compliant low frequency spectral component bit patterns as per section 6.7.7.1.1.3.
e) Compliant simultaneous switching outputs bit patterns as per section 6.7.7.1.1.4
f) Compliant composite patterns as per section 6.7.7.1.1.5

Where the qualifying prefix term "compliant" signifies transmission of the cited pattern encapsulated in the data portion of a valid Frame Information Structure, and used in a Serial ATA operational transmission context.

The Far-End Interface shall remain in this Far-End Retimed Loopback, until receipt of the COMRESET/COMINIT OOB Signaling sequence.

6.7.7.1.6.2 Test requirements - loopback - far-end analog (vendor specific)

The test requirements for this Loopback scheme, classified as Vendor Specific, are to be set by the Vendor.

6.7.7.1.6.3 Test requirements - loopback - near-end analog (vendor specific)

The test requirements for this Loopback scheme, classified as Vendor Specific, are to be set by the Vendor.

6.7.7.1.7 Test requirements - OOB signaling tests

Out-of-band signaling is used to signal specific actions during conditions where the receiving interface is in an active mode, a low interface power state, or a test mode.

This section specifies the set of test requirements to ensure that the OOB detector circuits will comply to the OOB signaling sequences under various conditions.

6.7.7.1.7.1.1 Power-on sequence

6.7.7.1.7.1.2 Calibration

When the host-controller performs impedance calibration, it shall adjust its own impedance such that the electrical requirements of section 6.6.2 are satisfied.

6.7.7.1.7.1.3 Speed negotiation

Speed Negotiation and transition to lower serial interface data rates shall be implemented for Gen2 compatible interfaces, negotiating and transitioning down to Gen1 speeds. There is no requirement for speed negotiation and transition to lower speeds than Gen1.

References to Gen1 and Gen2 speeds pertain to the definitions, per section 2.2.

6.7.7.1.7.2 Interface power management sequences

6.7.7.1.7.2.1 Partial

The interface shall detect the OOB signaling sequence "COMWAKE" and "COMRESET" when in the Partial Interface power management state.

While in the "Partial" state, the interface shall be subjected to the low-transition density bit pattern sequences of section 6.7.7.1.1; the interface shall remain in the Partial state until receipt of a valid COMWAKE (or COMRESET) OOB signaling sequence.

Power dissipation in this Partial state shall be measured or calculated to be less than the Phy Active State, but more than the Slumber State defined in section 6.8.

The requirement for a "not-to-exceed" power dissipation limit in the partial interface power management state will be classified as vendor specific, and will be documented as part of the implementation performance specifications.

6.7.7.1.7.2.2 Slumber

The interface shall detect the OOB signaling sequence "COMWAKE" and "COMRESET" when in the Slumber Interface power management state.

While in the "Slumber" state, the interface shall be subjected to the low-transition density bit pattern sequences of section 6.7.7.1.1; the interface shall remain in the Slumber state until receipt of a valid COMWAKE (or COMRESET) OOB signaling sequence.

Power dissipation in this Slumber state shall be measured or calculated to be less than the Phy Ready State, and less than the Partial State defined in section 6.8.

The requirement for a "not-to-exceed" power dissipation limit in the Slumber interface power management state will be classified as vendor specific, and will be documented as part of the implementation performance specifications.

6.7.7.2 Physical plant sub-modules

6.7.7.2.1 Receiver

6.7.7.2.1.1 Jitter tolerance

The requirements for receiver jitter tolerance have been implicitly addressed by other relevant sections.

6.7.7.2.1.2 Differential voltage

When subjected to the high-transition density patterns of section 6.7.7.1.1, the Differential Voltage measured at the Receiver shall comply to the electrical specifications of section 6.6.2.

6.7.7.2.1.3 Common-mode voltage

References to peak-to-peak voltages are cited in section 6.6.2.

The Receiver should operate to within the bit error rates cited in section 6.7.7.1.2, when subjected to a sinusoidal interfering signal with peak-to-peak voltage , and swept from the frequency range extremes, at a sweep rate period no shorter than 33.33 us.

The Receiver shall operate to within the frame error rates cited in section 6.7.7.1.3, when subjected to a sinusoidal interfering signal with peak-to-peak voltage and swept from the frequency range extremes, at a sweep rate period no shorter than 33.33 us.

6.7.7.2.2 Transmitter

6.7.7.2.2.1 Jitter output

Jitter output tests are intended to determine the jitter amplitudes of the random jitter, and deterministic jitter components.

6.7.7.2.2.1.1 Jitter measurements

The jitter specifications are based in a data transition to data transition Timing Interval Analyzer (TIA) format. If such a device is not available, equivalent measurements may be made with an oscilloscope. The oscilloscope is set up to trigger on the data and an EYE diagram is examined some integer number of unit intervals (UI) later. The histogram function can be used to monitor the distribution of the zero crossing to extract jitter information. Figure 43 shows an example where the oscilloscope is set up to examine an edge 16 UI away from the trigger point.

The DJ and RJ (or TJ) should be less than the maximum A_x indicated in Figure 43 – Jitter measurement example for n_x >= 16. For example, assume n_0 is five and n_1 is 250. In this case, the DJ and RJ for t_{16} must be less than A_1 since 250>16 but need not meet the A_0 requirement since 5<16. This method is employed because of the ease of measurement and to enable the specification to be relaxed at lower frequencies. If DJ is large enough, it can be fairly accurately measured from the histogram as the distance between the two outside peaks (there may be more than the three peaks shown in the example). If DJ and RJ are too intermixed for measurement, then other methods must be used. One such method is to transmit a clock pattern (fixed frequency components) to determine the RJ (DJ is zero for fixed frequencies), transmit a random patter to measure the TJ, and derive the DJ from these. Another alternative is to wait until $>>10^{12}$ edges are included in the histogram and measure the Total Jitter (TJ).

Since t_0 is triggered from the serial signal rather than a reference clock the resulting measurements do represent a combination of the jitter at t_0 and t_n.

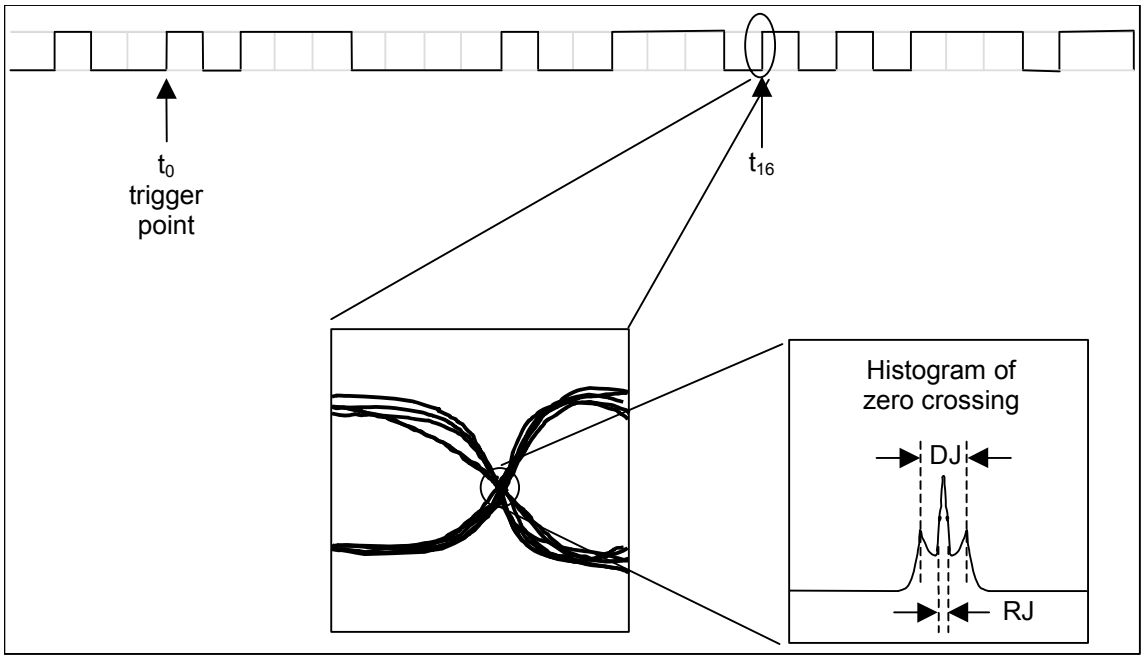


Figure 43 – Jitter measurement example

6.7.7.2.2.2 Differential voltage

When subjected to the high-transition density patterns of section 6.7.7.1.1, the Differential Voltage measured at the Transmitter shall comply to the respective electrical specifications of section 6.6.2.

6.7.7.2.2.3 Common-mode voltage

The Transmitter shall comply to the electrical specifications of section 6.6.2, when subjected to a sinusoidal interfering signal with peak-to-peak voltage, and swept from the frequency range extremes, at a sweep rate period no shorter than 33.33 us.

6.7.7.2.2.4 Rise/fall times

Output rise times and fall times are measured between 20% and 80% of the signal, see Signal Rise & Fall Times, Figure 44. Rise and fall time requirements apply to differential transitions, for both In-Band and Out-Of-Band signaling.

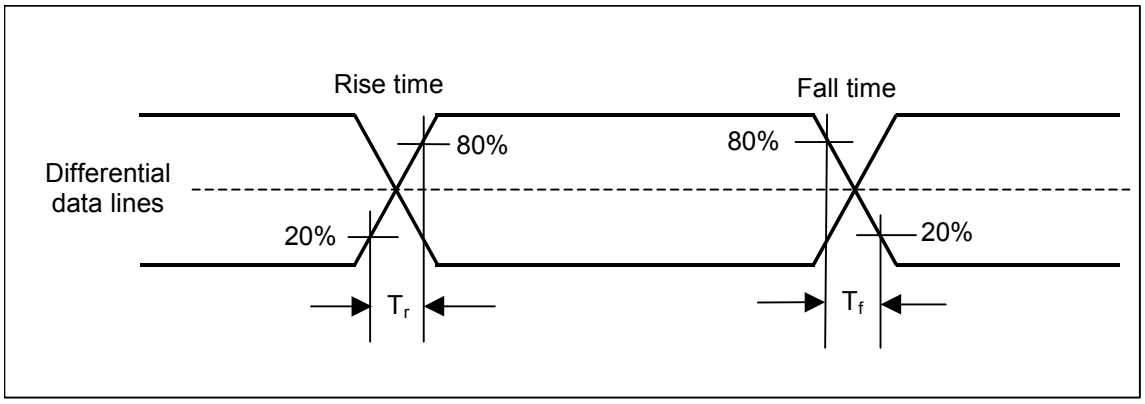


Figure 44 – Signal rise and fall times

The rise and fall times for transmitter differential buffer lines are measured with the load fixture shown in Figure 45, at the transmitter interface, and must comply to Electrical Specification paragraph 6.6.2 as well as be matched within ± 10% of each other to minimize RFI emissions and signal skew.

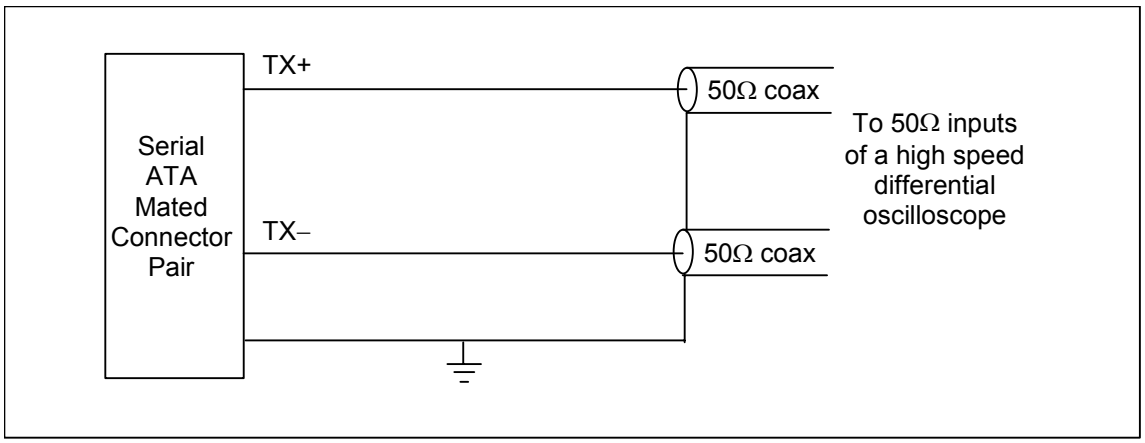


Figure 45 – Transmit test fixture

When testing the Serial ATA interface with the specified bit pattern sequences of section 6.7.7.1.1, use the Receiver Test fixture as shown in Figure 46.

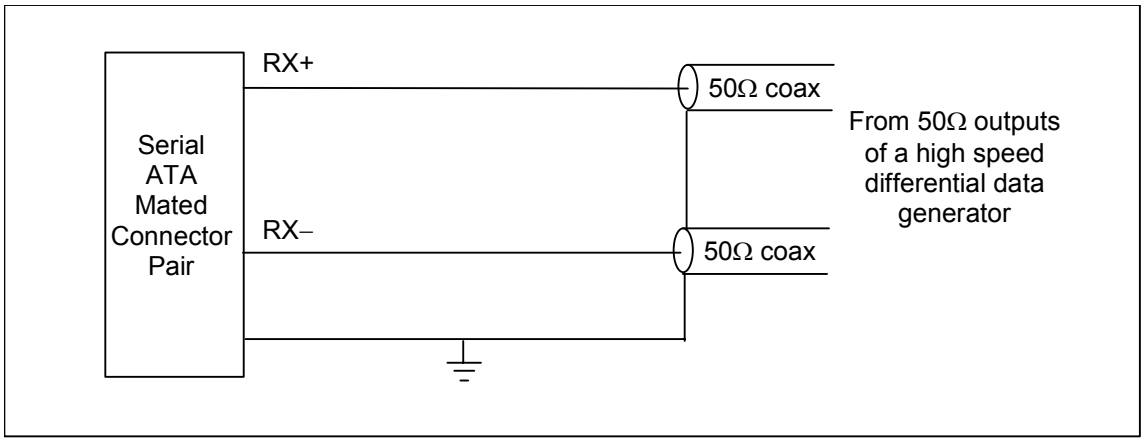


Figure 46 – Receive test fixture

6.8 Interface power states

Serial ATA Interface Power States will be controlled by the device and host controller. The Serial Interface Power States shall be defined as described in Table 15. – Interface Power States.

PHY READY	The PHY logic and main PLL are both on and active. The interface is synchronized and capable of receiving and sending data.
Partial	The PHY logic is powered, but is in a reduced power state. Both signal lines on the interface are at a neutral logic state (common mode voltage). The exit latency from this state shall be no longer than 10μs.
Slumber	The PHY logic is powered but is in a reduced power state. Both signal lines on the interface are at the neutral logic state (common mode voltage). The exit latency from this state shall be no longer than 10ms.

Table 15. – Interface Power States

6.8.1 Interface power state sequences

6.8.1.1 Power-on sequence state diagram

The following state diagrams specify the expected behavior of the host and device PHY from power-on to the establishment of an active communication channel.

In those states where the Phy relies on detection of received ALIGN primitives for state transitions, the Phy may rely on positive decode of the K28.5 leading character or the comma sequence (of either running disparity) as positive indication of ALIGN primitive reception rather than decoding the complete ALIGN primitive.

6.8.1.1.1 Host Phy Initialization state machine

As described in section 6.7.4.3, reception of a COMINIT signal shall cause the host to reinitialize communications with the device and shall unconditionally force the Host Phy state machine to transition to the HP2B:HR_AwaitNoCOMINIT state regardless of other conditions. Reception of COMINIT is effectively an additional transition into the HP2B:HR_AwaitNoCOMINIT state that appears in every Host Phy state. For the sake of brevity, this implied transition has been omitted from all the states.

HP1: HR_Reset	Transmit COMRESET[2, 3]	
1. Power-on reset and explicit reset request deasserted	→	HR_AwaitCOMINIT
2. Power-on reset or explicit reset request asserted	→	HR_Reset
NOTE : 1. This state is entered asynchronously any time in response to power-on reset or an explicit reset request. 2. Must transmit COMRESET for a minimum of 6 bursts (and a multiple of 6) 3. As described in section 6.7.4.2, COMRESET may be transmitted for the duration of this state, or it may be transmitted starting in this state and cease transmission after departure of this state, or it may be transmitted upon departure of this state.		

HP2: HR_AwaitCOMINIT	Interface quiescent	
1. COMINIT detected from device	→	HR_AwaitNoCOMINIT
2. COMINIT not detected from device	→	HR_AwaitCOMINIT

HP2B: HR_AwaitNoCOMINIT	Interface quiescent	
1. COMINIT not detected from device	→	HR_Calibrate
2. COMINIT detected from device	→	HR_AwaitNoCOMINIT
NOTE : 1. This state is entered asynchronously any time in response to COMINIT unless during a power-on reset or an explicit reset request (in which case HP1 is entered).		

HP3: HR_Calibrate	Perform calibration[1]		
	1. Calibration complete or bypass not implemented	→	HR_COMWAKE
	2. Calibration not complete	→	HR_Calibrate
	NOTE : 1. Calibration is optional. If bypassed or not implemented, proceed directly to HR_COMWAKE.		

HP4: HR_COMWAKE	Transmit COMWAKE		
	1. COMWAKE not detected from device	→	HR_AwaitCOMWAKE
	2. COMWAKE detected from device	→	HR_AwaitNoCOMWAKE

HP5: HR_AwaitCOMWAKE	Interface quiescent		
	1. COMWAKE detected from device	→	HR_AwaitNoCOMWAKE
	2. COMWAKE not detected from device	→	HR_AwaitCOMWAKE

HP5B: HR_AwaitNoCOMWAKE	Interface quiescent		
	1. COMWAKE not detected from device	→	HR_AwaitAlign
	2. COMWAKE detected from device	→	HR_AwaitNoCOMWAKE

HP6: HR_AwaitAlign	Host transmits D10.2 characters at lowest supported rate[2]		
	1. ALIGN detected from device (at any supported speed)[3]	→	HR_AdjustSpeed
	2. ALIGN not detected from device and 880us (32768 Gen1 dwords) has elapsed since entry to HR_AwaitAlign	→	HR_Reset[1,4]
	3. ALIGN not detected from device and less than 880us (32768 Gen1 dwords) has elapsed since entry to HR_AwaitAlign	→	HR_AwaitAlign
	NOTE : 1. Host retries the power-on sequence indefinitely unless explicitly turned off by the application layer 2. Host must start transmitting d10.2 characters no later than 533ns (20 Gen1 dwords) after COMWAKE is deasserted as specified in the out of band signaling section 3. Host designers should be aware that the device is allowed 53.3ns (2 Gen1 dwords) after releasing COMWAKE (by holding the idle condition for more than 175ns) to start sending characters. Until this occurs, the bus will be at an idle condition and may be susceptible to crosstalk from other devices. Care must be taken so that crosstalk during this window doesn't result in a false detection of an ALIGN. For example: a compliant host may detect the deassertion of COMWAKE in as little as 112ns, such a host should wait at least 116.3ns (175+53.3-112) after detecting the release of COMWAKE to start looking for ALIGNs. 4. The Host PHY state machine may use the transition to HR_Reset as a method of speed negotiation.		

HP7: HR_SendAlign	Transmit ALIGN at speed detected	
1. Three back-to-back non-ALIGN primitives[2] detected from device	→	HR_Ready
2. Three back-to-back non-ALIGN primitives not detected from device	→	HR_SendAlign[1]
NOTE : 1. Host retries indefinitely unless explicitly turned off by the application layer 2. Non-ALIGN primitives can be detected by the presence of the k28.3 control character in the byte0 position		

HP8: HR_Ready	Transmit word from Link[1]	
1. Partial signal from Link asserted	→	HR_Partial
2. Slumber signal from Link asserted	→	HR_Slumber
3. No power management request received	→	HR_Ready
NOTE : 1. PhyRdy asserted only when in the HR_Ready state and the Phy is maintaining synchronization with the incoming signal to its receiver and is transmitting a valid signal on its transmitter.		

HP9: HR_Partial	Interface quiescent	
1. Partial signal from Link deasserted and no COMWAKE detected from device[1]	→	HR_COMWAKE
2. Partial signal from Link deasserted and COMWAKE detected from device[1]	→	HR_AwaitNoCOMWAKE
3. Partial signal from Link asserted	→	HR_Partial
NOTE : 1. Host Phy must remember if COMWAKE was detected during Partial to determine if the wakeup request originated from the host or the Phy.		

HP10: HR_Slumber	Interface quiescent	
1. Slumber signal from Link deasserted and no COMWAKE detected from device[1,2]	→	HR_COMWAKE
2. Slumber signal from Link deasserted and COMWAKE detected from device[1,2]	→	HR_AwaitNoCOMWAKE
3. Slumber signal from Link asserted	→	HR_Slumber

NOTE :
1. Host Phy must remember if COMWAKE was detected during Slumber to determine if the wakeup request originated from the host or the Phy.
2. The host Phy may take this transition only after it has recovered from slumber mode and the Phy is prepared to initiate communications. If Phy has not yet recovered from the slumber mode it shall remain in this state.

HP11: HR_AdjustSpeed	Interface undefined but not quiescent[1]	
1 Transition to appropriate speed completed	→	HR_SendAlign
2 Transition to appropriate speed not completed	→	HR_AdjustSpeed

NOTE :
1. Some implementations may undergo a transient condition where invalid signals are transmitted during the change in their internal transmission/reception speed. The host may transmit invalid signals for a period of up to 53ns (two Gen1 dwords) during the speed transition. Transmit jitter and unit interval timing requirements may not be met during this period but shall be met for all other bits transmitted in this state. A Phase shift may occur across the speed transition time.

6.8.1.1.2 Device Phy Initialization state machine

As described in section 6.7.4.2, reception of a COMRESET signal shall be treated by the device as a hard reset signal and shall unconditionally force the Device Phy state machine to transition to the DP1:DR_Reset initial state regardless of other conditions. Reception of COMRESET is effectively an additional transition into the DP1:DR_Reset state that appears in every Device Phy state. For the sake of brevity, this implied transition has been omitted from all the states.

DP1: DR_Reset[1]	Interface quiescent	
1. COMRESET not detected and power-on reset deasserted	→	DR_COMINIT
2. COMRESET detected or power-on reset asserted	→	DR_Reset
NOTE : 1. This state is entered asynchronously any time in response to power-on reset or receipt of a COMRESET signal from the host		

DP2: DR_COMINIT	Transmit COMINIT[1]	
1. Unconditional	→	DR_AwaitCOMWAKE
NOTE : 1. COMINIT transmitted for a 6 bursts duration		

DP3: DR_AwaitCOMWAKE	Interface quiescent	
1. COMWAKE detected from host	→	DR_AwaitNoCOMWAKE
2. COMWAKE not detected from host	→	DR_AwaitCOMWAKE

DP3B: DR_AwaitNoCOMWAKE	Interface quiescent	
1. COMWAKE not detected from host and part of power-on reset sequence[1]	→	DR_Calibrate
2. COMWAKE not detected from host and part of partial/slumber awake sequence[1]	→	DR_COMWAKE
3. COMWAKE detected from host	→	DR_AwaitNoCOMWAKE
NOTE : 1. Device must remember if it was sent to partial or slumber mode for proper wakeup action.		

DP4: DR_Calibrate	Perform calibration[1]	
1. Calibration complete or bypass not implemented	→	DR_COMWAKE
2. Calibration not complete	→	DR_Calibrate
NOTE : 1. Calibration is optional. If bypassed or not implemented, proceed directly to DR_COMWAKE.		

DP5: DR_COMWAKE	Transmit COMWAKE	
1. Unconditional	→	DR_SendAlign

DP6: DR_SendAlign	Transmit ALIGN[1,2,3,5]	
1. ALIGN detected from host (device locked to incoming data)[4]	→	DR_Ready
2. ALIGN not detected from host and ALIGN primitives transmitted for 54.6us (2048[5] Gen1 ALIGN primitives) at speed other than lowest[6]	→	DR_ReduceSpeed
3. ALIGN not detected from host and ALIGN primitives transmitted for 54.6us (2048[5] Gen1 ALIGN primitives) at lowest speed[6]	→	DR_Error
4. ALIGN not detected from host and ALIGN primitives transmitted for less than 54.6us (2048 Gen1 ALIGN primitives)	→	DR_SendAlign

NOTE :
1. ALIGN should be sent at the devices fastest supported speed first
2. ALIGNS should be sent only at valid frequencies (if PLL not locked, send D10.2)
3. After COMWAKE is released as specified in the out of band signaling section, the device must ensure the interface is active (not quiescent).
4. Device designers should be aware that the host is allowed 533ns (20 Gen1 dwords) after detecting the deassertion of COMWAKE to start sending d10.2 characters. Until this occurs, the bus will be at an idle condition and may be susceptible to crosstalk from other devices. Care must be taken so that crosstalk during this window doesn't result in a false detection of an ALIGN. Devices may extend this timeout up to an additional 54.6us (2048 Gen1 dwords) (for a max total of 109.2us), as necessary to allow their receiver time to lock to the host ALIGN.
5. Device must not leave the bus idle more than 53.3ns (2 Gen1 dwords) longer than the required 175ns to deassert COMWAKE.
6. If this is part of a device-initiated recovery from the Slumber or Partial power management state, the device Phy should resume at the speed previously negotiated and should not reduce its speed in response to failure to establish communications. Upon failing to establish communications it should instead transition directly to the DR_Error state to initiate a re-try of the ComWake sequence.

DP7: DR_Ready[1]	Transmit word from Link	
1. Partial signal from Link asserted	→	DR_Partial
2. Slumber signal from Link asserted	→	DR_Slumber
3. No power management request received	→	DR_Ready

NOTE :
1. PhyRdy asserted only when in the DR_Ready state and the Phy is maintaining synchronization with the incoming signal to its receiver and is transmitting a valid signal on its transmitter.

DP8: DR_Partial	Interface quiescent	
1. Partial signal from Link deasserted	→	DR_COMWAKE
2. Partial signal from Link deasserted and COMWAKE detected from host	→	DR_AwaitNoCOMWAKE
3. Partial signal from Link asserted	→	DR_Partial

DP9: DR_Slumber	Interface quiescent	
1. Slumber signal from Link deasserted	→	DR_COMWAKE
2. Slumber signal from Link deasserted and COMWAKE detected from host	→	DR_AwaitNoCOMWAKE
3. Slumber signal from Link asserted	→	DR_Slumber

DP10: DR_ReduceSpeed	Interface quiescent	
1. Transition to legacy (slower) speed complete	→	DR_SendAlign[1]
2. Transition to legacy speed not complete	→	DR_ReduceSpeed
NOTE : 1. Transition to a new speed is defined as being complete when the device is accurately transmitting a valid signal within the defined signaling tolerances for that speed.		

DP11: DR_Error	Interface quiescent	
1. Error not due to failure to resume	→	DR_Error
2. Resume from Slumber or Partial failed	→	DR_COMWAKE

6.8.1.2 Power-on sequence timing diagram

The following timing diagrams and descriptions are provided for clarity and are informative. The state diagrams provided in section 6.8.1.1 comprise the normative behavior specification and is the ultimate reference.

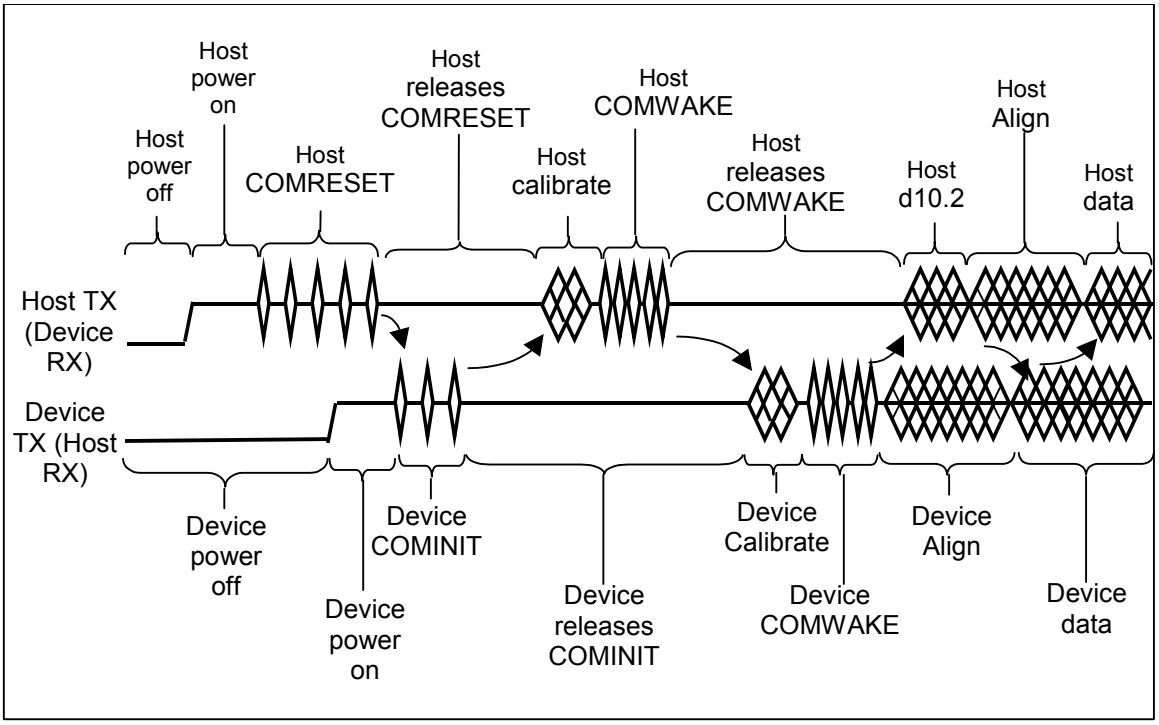


Figure 47 – Power-on Sequence

Description:

1. Host/device power-off - Host and device power-off. There is host side signal conditioning that will pull the host TX and RX pairs low if power is off.
2. Power is applied - Host side signal conditioning pulls TX and RX pairs to neutral state (common mode voltage).
3. Host issues COMRESET
4. Host releases COMRESET. Once the power-on reset is released, the host releases the COMRESET signal and puts the bus in a quiescent condition.
5. Device issues COMINIT – When the device detects the release of COMRESET, it responds with a COMINIT. This is also the entry point if the device is late starting. The device may initiate communications at any time by issuing a COMINIT.
6. Host calibrates and issues a COMWAKE.
7. Device responds – The device detects the COMWAKE signal on its RX pair and calibrates its transmitter (optional). Following calibration, the device sends a six burst COMWAKE signal and then sends a continuous stream of the ALIGN sequence. After ALIGN primitives have been sent for 54.6us (2048 Gen1 ALIGN Dwords) without a valid response from the host as determined by detection of received ALIGN primitives , the device may assume that the host cannot communicate at that speed. If additional legacy speeds are available, the device will try the next slower speed by sending ALIGN primitives for 54.6us at that rate. This step is repeated for as many legacy speeds are supported. Once the lowest speed has been reached without response from the host, the device will enter an error state.
8. Host locks – after detecting the COMWAKE, the host starts transmitting D10.2 characters (see 6.7.6) at its lowest supported rate. Meanwhile, the host receiver locks to the ALIGN sequence and, when ready, returns the ALIGN sequence to the device at the same speed as received. A host must be designed such that it can acquire lock within 54.6us (2048 Gen1 ALIGN Dwords). The host should wait for at least 880us (32768 Gen1 dwords) after detecting the release of COMWAKE to receive the first ALIGN. This will ensure interoperability with multi-generational and synchronous devices. If no ALIGN is received within 880us (32768 Gen1 dwords), the host shall restart the power-on sequence – repeating indefinitely until stopped by the application layer.
9. Device locks – the device locks to the ALIGN sequence and, when ready, sends the SYNC primitive indicating it is ready to start normal operation.
10. Upon receipt of three back-to-back non-ALIGN primitives, the communication link is established and normal operation may begin.

6.8.1.3 Partial/Slumber to on

6.8.1.3.1 Host initiated

The host can initiate a wakeup from the partial or slumber states by entering the power-on sequence at the “Host COMWAKE” point in the state machine. Calibration and speed negotiation is bypassed since it has already been performed at power-on and system performance depends on quick resume latency. The device, therefore, shall transmit ALIGNs at the speed determined at power-on.

6.8.1.3.2 Device initiated

The device can initiate a wakeup from the partial or slumber states by entering the power-on sequence at the “Device COMWAKE” point in the state machine. Calibration and speed negotiation is bypassed since it has already been performed at power-on and system performance depends on quick resume latency. The device, therefore, shall transmit ALIGNs at the speed determined at power-on.

6.8.1.4 On to Partial/Slumber

6.8.1.4.1 Host initiated

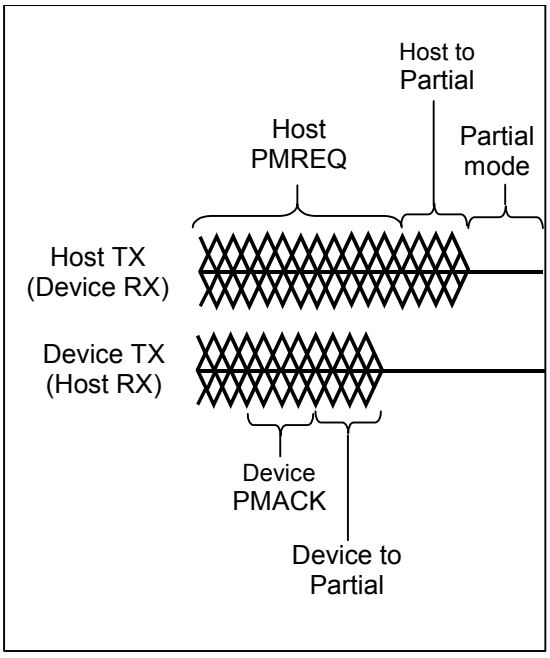


Figure 48 – On to Partial/Slumber - host initiated

6.8.1.4.1.1 Detailed sequence

1. Host Application layer sends request to host Transport layer.
2. Host Transport layer transmits request to host Link layer.
3. Host Link layer encodes request as PMREQ primitive and transmits it four times to host Phy layer.
4. Host Phy layer serializes PMREQ primitives and transmits them to device Phy layer.
5. Device Phy de-serializes PMREQ primitives and transmits them to device Link layer.
6. Device Link layer decodes PMREQ primitives and transmits request to device Transport layer.
7. Device Transport layer transmits request to device Application layer.
8. Device Application layer processes and accepts request. Issues accept to device Transport layer.
9. Device Transport layer transmits acceptance to device Link layer.
10. Device Link layer encodes acceptance as PMACK primitive and transmits it four times to device Phy layer.
11. Device Phy layer transmits four PMACK primitives to host Phy layer.
12. Device Link layer places device Phy layer in Partial/Slumber state.
13. Host Phy layer de-serializes PMACK primitives and transmits them to host Link layer.
14. Host Link layer decodes PMACK primitives and transmits acceptance to host Transport layer.
15. Host Link layer places host Phy layer in Partial/Slumber State.
16. Host Transport layer transmits acceptance to host Application layer.

6.8.1.4.2 Device initiated

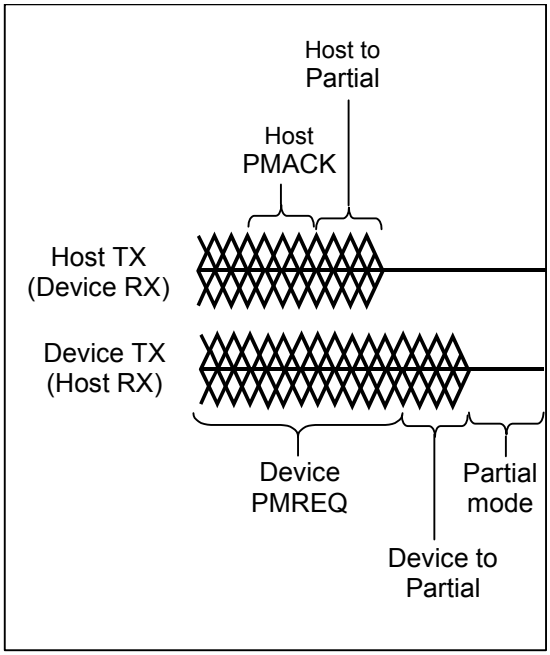


Figure 49 – ON to Partial/Slumber - device initiated

6.8.1.4.2.1 Detailed sequence

1. Device Application layer sends request to device Transport layer.
2. Device Transport layer transmits request to device Link layer.
3. Device Link layer encodes request as PMREQ primitive and transmits it to device Phy layer.
4. Device Phy layer serializes PMREQ primitives and transmits them to host Phy layer.
5. Host Phy de-serializes PMREQ primitives and transmits them to host Link layer.
6. Host Link layer decodes PMREQ primitives and transmits request to host Transport layer.
7. Host Transport layer transmits request to host Application layer.
   NOTE – In this context, the host Application layer does not necessarily imply BIOS or other host CPU programming. Rather, the Application layer is the intelligent control section of the chipset logic.
8. Host Application layer processes and accepts request. Issues accept to host Transport layer.
9. Host Transport layer transmits acceptance to host Link layer.
10. Host link layer encodes acceptance as PMACK primitive and transmits it four times to host Phy layer.
11. Host Phy layer transmits four PMACK primitives to device Phy layer.
12. Host Link layer asserts Partial/Slumber signal and places host Phy layer in Partial/Slumber state.
13. Host Phy layer negates Ready signal.
14. Device Phy layer de-serializes PMACK primitives and transmits them to device Link layer.
15. Device Link layer decodes PMACK primitives and transmits acceptance to device Transport layer.
16. Device Link layer asserts Partial/Slumber signal and places device Phy layer in Partial/Slumber State.
17. Device Phy layer negates Ready signal.
18. Device Transport layer transmits acceptance to device Application layer.

7 Link layer

7.1.1 Overview

The Link layer transmits and receives frames, transmits primitives based on control signals from the Transport layer, and receives primitives from the Phy layer which are converted to control signals to the Transport layer. The Link layer need not be cognizant of the content of frames. Host and device Link layer state machines differ only in the fact that it is assumed that the host shall back off when transmission buffer is full.

7.1.1.1 Frame transmission

When requested by the Transport layer to transmit a frame, the Link layer provides the following services:

- Negotiates with its peer Link layer to transmit a frame, resolves arbitration conflicts if both host and device request transmission
- Inserts frame envelope around Transport layer data (i.e., SOF, CRC, EOF, etc.).
- Receives data in the form of Dwords from the Transport layer.
- Calculates CRC on Transport layer data.
- Transmits frame.
- Provides frame flow control in response to requests from the FIFO or the peer Link layer.
- Receives frame receipt acknowledge from peer Link layer.
- Reports good transmission or Link/Phy layer errors to Transport layer.
- Performs 8b/10b encoding
- Transforms (scrambles) control and data Dwords in such a way to distribute the potential EMI emissions over a broader range

7.1.1.2 Frame receipt

When data is received from the Phy layer, the Link layer provides the following services:

- Acknowledges to the peer Link layer readiness to receive a frame.
- Receives data in the form of encoded characters from the Phy layer.
- Decodes the encoded 8b/10b character stream into aligned Dwords of data.
- Removes the envelope around frames (i.e., SOF, CRC, EOF).
- Calculates CRC on the received Dwords.
- Provides frame flow control in response to requests from the FIFO or the peer Link layer.
- Compares the calculated CRC to the received CRC.
- Reports good reception or Link/Phy layer errors to Transport layer and the peer Link layer.
- Untransforms (descrambles) the control and data Dwords received from a peer Link layer.

7.2 Encoding method

Information to be transmitted over the Serial ATA bus shall be encoded a byte (eight bits) at a time along with a data or control character indicator into a 10-bit encoded character and then sent serially bit by bit. Information received over the Serial ATA bus shall be collected ten bits at a time, assembled into an encoded character, and decoded into the correct data characters and control characters. The 8b/10b code allows for the encoding of all 256 combinations of eight-bit data. A smaller subset of the control character set is utilized by Serial ATA.

7.2.1 Notation and conventions

Serial ATA uses a letter notation for describing data bits and control variables. A description of the translation process between these notations follows. This section also describes a convention used

to differentiate data characters from control characters. Finally, translation examples for both a data character and a control character are presented.

An unencoded byte of data is composed of eight bits A,B,C,D,E,F,G,H and the control variable Z. The encoding process results in a 10 bit character a,b,c,d,e,i,f,g,h,j. A bit is either a binary zero or binary one. The control variable, Z, has a value of D or K. When the control variable associated with a byte has the value D, the byte is referred to as a data character. When the control variable associated with a byte has the value K, the byte is referred to as a control character.

If a data byte is not accompanied with a specific control variable value the control variable Z is assumed to be Z = D and the data byte shall be encoded as a data character.

The following figure illustrates the association between the numbered unencoded bits in a byte, the control variable, and the letter-labeled bits in the encoding scheme:

Data byte notation	7	6	5	4	3	2	1	0	Control variable
Unencoded bit notation	H	G	F	E	D	C	B	A	Z

Figure 50 – Bit designations

Each character is given a name Zxx.y where Z is the value of the control variable (D for a data character, K for a control character), xx is the decimal value of the binary number composed of the bits E, D, C, B and A in that order, and y is the decimal value of the binary number composed of the bits H, G and F.

Figure 51, following, shows the relationship between the various representations.

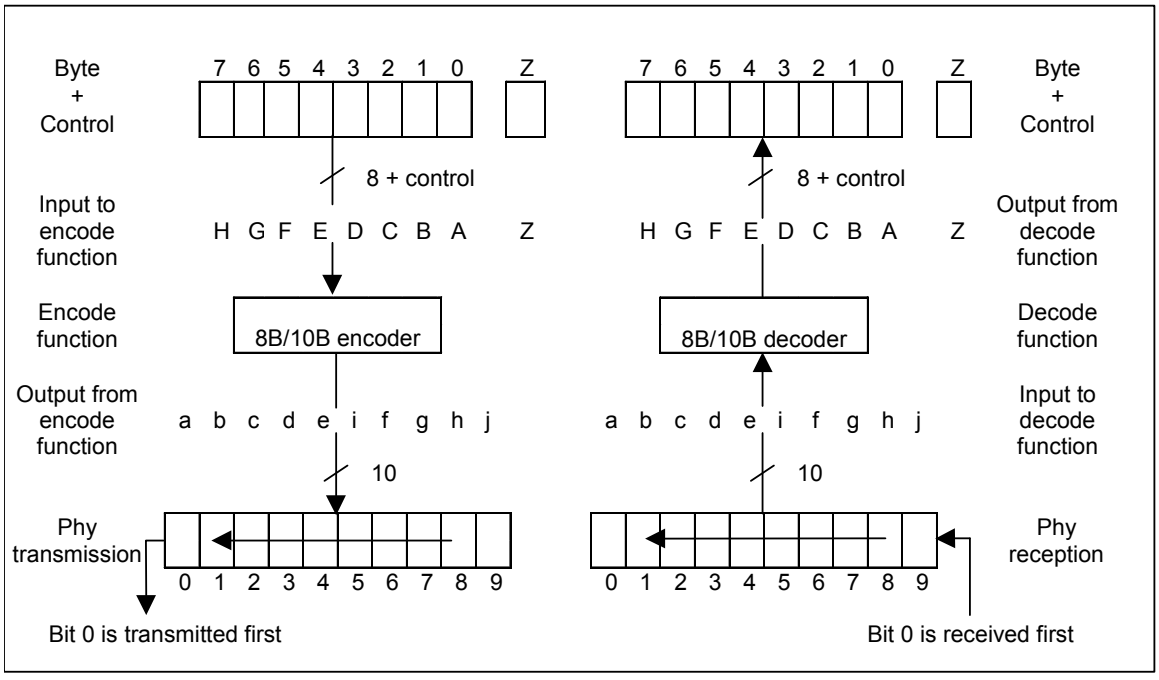


Figure 51 – Nomenclature reference

Figure 52 shows conversions from byte notation to character notation for a control and data byte. The examples chosen have special significance and will also be used during the conversion from data notation to the 8b/10b code values.

Byte notation	BCh, control character		4Ah, data character	
Bit notation	76543210 10111100	Control variable K	76543210 01001010	Control variable D
Unencoded bit notation	HGF EDCBA 101 11100	Z K	HGF EDCBA 010 01010	Z D
Bit notation reordered to conform with Zxx.y convention	Z EDCBA HGF K 11100 101		Z EDCBA HGF D 01010 010	
Character name	K 28 .5		D 10 .2	

Figure 52 – Conversion examples

7.2.2 Character code

The coding scheme used by Serial ATA translates unencoded data and control bytes to characters. The encoded characters are then transmitted by the physical layer over the serial line where they are received from the physical layer and decoded into the corresponding byte and control value.

Serial ATA uses a subset of the 8b/10b coding method described by Widmer and Franaszek. [1] [2] The Serial ATA code uses all 256 data byte encodings while only two of the control codes are used. The reception of any unused code is a class of reception error referred to as a code violation.

7.2.2.1 Code construction

The 8b/10b coding process is defined in two stages. The first stage encodes the first five bits of the unencoded input byte into a six bit sub-block using a 5B/6B encoder. The input to this stage includes the current running disparity value. The second stage uses a 3B/4B encoder to encode the remaining three bits of the data byte and the running disparity as modified by the 5B/6B encoder into a four bit value.

In the derivations that follow, the control variable (Z) is assumed to have a value of D, and thus is an implicit input.

7.2.2.2 The concept of running disparity

Running Disparity is a binary parameter with either the value negative (-) or the value positive (+).

After transmitting any encoded character, the transmitter shall calculate a new value for its Running Disparity based on the value of the transmitted character.

After a COMRESET, initial power-up, exiting any power management state, or exiting any diagnostic mode, the receiver shall assume either the positive or negative value for its initial Running Disparity. Upon reception of an encoded character the receiver shall determine whether the encoded character is valid according to the following rules and tables and shall calculate a new value for its Running Disparity based on the contents of the received character.

The following rules shall be used to calculate a new Running Disparity value for the transmitter after it sends an encoded character (transmitter's new Running Disparity) and for the receiver upon reception of an encoded character (receiver's new Running Disparity).

Running Disparity for an encoded character shall be calculated on two sub-blocks where the first six bits (abcdei) form one sub-block – the six-bit sub-block. The last four bits (fghj) form the second sub-block – the four-bit sub-block. Running Disparity at the beginning of the six-bit sub-block is the Running Disparity at the end of the last encoded character or the initial conditions described above for the first encoded character transmitted or received. Running Disparity at the beginning of the four-bit sub-block is the resulting Running Disparity from the six-bit sub-block. Running Disparity at the end of the encoded character – and the initial Running Disparity for the next encoded character – is the Running Disparity at the end of the four-bit sub-block.

Running Disparity for each of the sub-blocks shall be calculated as follows:

Running Disparity at the end of any sub-block is positive if the sub-block contains more ones than zeros. It is also positive at the end of the six-bit sub-block if the value of the six-bit sub-block is 000111, and is positive at the end of the four-bit sub-block if the value of the four-bit sub-block is 0011.

Running Disparity at the end of any sub-block is negative if the sub-block contains more zeros than ones. It is also negative at the end of the six-bit sub-block if the value of the six-bit sub-block is 111000, and is negative at the end of the four-bit sub-block if the value of the four-bit sub-block is 1100.

Otherwise, for any sub-block with an equal number of zeros and ones, the Running Disparity at the end of the sub-block is the same as at the beginning of the sub-block. Sub-blocks with an equal number of zeros and ones are said to have neutral disparity.

The 8b/10b code restricts the generation of the 000111, 111000, 0011 and 1100 sub-blocks in order to limit the run length of zeros and ones between sub-blocks. Sub-blocks containing 000111 or 0011 are generated only when the running disparity at the beginning of the sub-block is positive, resulting in positive Running Disparity at the end of the sub-block. Similarly, sub-blocks containing 111000 or 1100 are generated only when the running disparity at the beginning of the sub-block is negative and the resulting Running Disparity will also be negative.

The rules for Running Disparity will result in generation of a character with disparity that is either the opposite of the previous character or neutral.

Sub-blocks with non-zero (non-neutral) disparity must be of alternating disparity.

7.2.2.3 Data encoding

Table 16 and Table 17 describe the code and running disparity generation rules for each of the sub-blocks. The results can be used to generate the data in the data character tables.

The digital logic which is used to generate the results can be found in Franaszek and Widmer [2]. The generation of control characters is also covered in the patent but not here.

In the tables which follow rd+ or rd- represent the current (incoming) running disparity and rd' represents the resulting Running Disparity. The resulting Running Disparity columns use –rd to indicate a change in Running Disparity polarity while rd indicates the resulting sub-block has neutral disparity.

Table 16. – 5B/6B coding

Inputs		abcdei outputs		rd'	Inputs		abcdei outputs		rd'
Dx	EDCBA	rd+	rd-		Dx	EDCBA	rd+	rd-	
D0	00000	011000	100111	-rd	D16	10000	100100	011011	-rd
D1	00001	100010	011101		D17	10001	100011		rd
D2	00010	010010	101101		D18	10010	010011		
D3	00011	110001		rd	D19	10011	110010		
D4	00100	001010	110101	-rd	D20	10100	001011		
D5	00101	101001		rd	D21	10101	101010		
D6	00110	011001			D22	10110	011010		
D7	00111	000111	111000		D23	10111	000101	111010	-rd
D8	01000	000110	111001	-rd	D24	11000	001100	110011	
D9	01001	100101		rd	D25	11001	100110		rd
D10	01010	010101			D26	11010	010110		
D11	01011	110100			D27	11011	001001	110110	-rd
D12	01100	001101			D28	11100	001110		rd
D13	01101	101100			D29	11101	010001	101110	-rd
D14	01110	011100			D30	11110	100001	011110	
D15	01111	101000	010111	-rd	D31	11111	010100	101011	

Table 17. – 3B/4B coding

Inputs		fghj outputs		rd'
Dx.y	HGF	rd+	rd-	
Dx.0	000	0100	1011	-rd
Dx.1	001	1001		rd
Dx.2	010	0101		
Dx.3	011	0011	1100	
Dx.4	100	0010	1101	-rd
Dx.5	101	1010		rd
Dx.6	110	0110		
Dx.P7	111	0001	1110	-rd
Dx.A7	111	1000	0111	

NOTE –
A7 replaces P7 if[(rd>0) and (e=i=0)] or [(rd<0) and (e=i=1)]

7.2.2.4 Encoding examples

The coding examples in Figure 53 illustrate how the running disparity calculations are done.

The first conversion example completes the translation of data byte value 4Ah (which is the character name of D10.2) into an encoded character value of “abcdei fghj” = “010101 0101”. This value has special significance because (1) it is of neutral disparity, and also contains an alternating zero/one pattern that represents the highest data frequency which can be generated.

In the second example the 8b/10b character named D11.7 is encoded. Assuming a positive value for the incoming Running Disparity, this example shows the Dx.P7/Dx.A7 substitution. With an initial rd+ value, D10 translates to an abcdei value of 110100, with a resulting Running Disparity of positive for the 6-bit sub-block. Encoding the 4-bit sub-block will trigger the substitution clause of Dx.A7 for Dx.P7 since [(rd>0) AND (e=i=0)].

Initial rd	Character name	abcdei output	6-bit sub-block rd	fghj output	4-bit sub-block rd	Encoded character	Ending rd
-	D10.2	010101	-	0101	-	010101 0101	-
+	D11.7	110100	+	1000	-	110100 1000	-

Figure 53 – Coding examples

7.2.2.5 8b/10b valid encoded characters

The following tables define the valid data characters and valid control characters. These tables shall be used for generating encoded characters (encoding) for transmission. In the reception process, the table is used to look up and verify the validity of received characters (decoding).

In the tables, each data character and control character has two columns that represent two encoded characters. One column represents the output if the current Running Disparity is negative and the other is the output if the current Running Disparity is positive.

7.2.2.5.1 Data characters

Table 18. – Valid data characters

Name	Byte	abcdei fghj output		Name	Byte	abcdei fghj output	
		Current rd-	Current rd+			Current rd-	Current rd+
D0.0	00h	100111 0100	011000 1011	D0.1	20h	100111 1001	011000 1001
D1.0	01h	011101 0100	100010 1011	D1.1	21h	011101 1001	100010 1001
D2.0	02h	101101 0100	010010 1011	D2.1	22h	101101 1001	010010 1001
D3.0	03h	110001 1011	110001 0100	D3.1	23h	110001 1001	110001 1001
D4.0	04h	110101 0100	001010 1011	D4.1	24h	110101 1001	001010 1001
D5.0	05h	101001 1011	101001 0100	D5.1	25h	101001 1001	101001 1001
D6.0	06h	011001 1011	011001 0100	D6.1	26h	011001 1001	011001 1001
D7.0	07h	111000 1011	000111 0100	D7.1	27h	111000 1001	000111 1001
D8.0	08h	111001 0100	000110 1011	D8.1	28h	111001 1001	000110 1001
D9.0	09h	100101 1011	100101 0100	D9.1	29h	100101 1001	100101 1001
D10.0	0Ah	010101 1011	010101 0100	D10.1	2Ah	010101 1001	010101 1001
D11.0	0Bh	110100 1011	110100 0100	D11.1	2Bh	110100 1001	110100 1001
D12.0	0Ch	001101 1011	001101 0100	D12.1	2Ch	001101 1001	001101 1001
D13.0	0Dh	101100 1011	101100 0100	D13.1	2Dh	101100 1001	101100 1001
D14.0	0Eh	011100 1011	011100 0100	D14.1	2Eh	011100 1001	011100 1001
D15.0	0Fh	010111 0100	101000 1011	D15.1	2Fh	010111 1001	101000 1001
D16.0	10h	011011 0100	100100 1011	D16.1	30h	011011 1001	100100 1001
D17.0	11h	100011 1011	100011 0100	D17.1	31h	100011 1001	100011 1001
D18.0	12h	010011 1011	010011 0100	D18.1	32h	010011 1001	010011 1001
D19.0	13h	110010 1011	110010 0100	D19.1	33h	110010 1001	110010 1001
D20.0	14h	001011 1011	001011 0100	D20.1	34h	001011 1001	001011 1001
D21.0	15h	101010 1011	101010 0100	D21.1	35h	101010 1001	101010 1001
D22.0	16h	011010 1011	011010 0100	D22.1	36h	011010 1001	011010 1001
D23.0	17h	111010 0100	000101 1011	D23.1	37h	111010 1001	000101 1001
D24.0	18h	110011 0100	001100 1011	D24.1	38h	110011 1001	001100 1001
D25.0	19h	100110 1011	100110 0100	D25.1	39h	100110 1001	100110 1001
D26.0	1Ah	010110 1011	010110 0100	D26.1	3Ah	010110 1001	010110 1001
D27.0	1Bh	110110 0100	001001 1011	D27.1	3Bh	110110 1001	001001 1001
D28.0	1Ch	001110 1011	001110 0100	D28.1	3Ch	001110 1001	001110 1001
D29.0	1Dh	101110 0100	010001 1011	D29.1	3Dh	101110 1001	010001 1001
D30.0	1Eh	011110 0100	100001 1011	D30.1	3Eh	011110 1001	100001 1001
D31.0	1Fh	101011 0100	010100 1011	D31.1	3Fh	101011 1001	010100 1001

(continued)

Table 18 – Valid data characters *(continued)*

Name	Byte	abcdei fghj output		Name	Byte	abcdei fghj output	
		Current rd-	Current rd+			Current rd-	Current rd+
D0.2	40h	100111 0101	011000 0101	D0.3	60h	100111 0011	011000 1100
D1.2	41h	011101 0101	100010 0101	D1.3	61h	011101 0011	100010 1100
D2.2	42h	101101 0101	010010 0101	D2.3	62h	101101 0011	010010 1100
D3.2	43h	110001 0101	110001 0101	D3.3	63h	110001 1100	110001 0011
D4.2	44h	110101 0101	001010 0101	D4.3	64h	110101 0011	001010 1100
D5.2	45h	101001 0101	101001 0101	D5.3	65h	101001 1100	101001 0011
D6.2	46h	011001 0101	011001 0101	D6.3	66h	011001 1100	011001 0011
D7.2	47h	111000 0101	000111 0101	D7.3	67h	111000 1100	000111 0011
D8.2	48h	111001 0101	000110 0101	D8.3	68h	111001 0011	000110 1100
D9.2	49h	100101 0101	100101 0101	D9.3	69h	100101 1100	100101 0011
D10.2	4Ah	010101 0101	010101 0101	D10.3	6Ah	010101 1100	010101 0011
D11.2	4Bh	110100 0101	110100 0101	D11.3	6Bh	110100 1100	110100 0011
D12.2	4Ch	001101 0101	001101 0101	D12.3	6Ch	001101 1100	001101 0011
D13.2	4Dh	101100 0101	101100 0101	D13.3	6Dh	101100 1100	101100 0011
D14.2	4Eh	011100 0101	011100 0101	D14.3	6Eh	011100 1100	011100 0011
D15.2	4Fh	010111 0101	101000 0101	D15.3	6Fh	010111 0011	101000 1100
D16.2	50h	011011 0101	100100 0101	D16.3	70h	011011 0011	100100 1100
D17.2	51h	100011 0101	100011 0101	D17.3	71h	100011 1100	100011 0011
D18.2	52h	010011 0101	010011 0101	D18.3	72h	010011 1100	010011 0011
D19.2	53h	110010 0101	110010 0101	D19.3	73h	110010 1100	110010 0011
D20.2	54h	001011 0101	001011 0101	D20.3	74h	001011 1100	001011 0011
D21.2	55h	101010 0101	101010 0101	D21.3	75h	101010 1100	101010 0011
D22.2	56h	011010 0101	011010 0101	D22.3	76h	011010 1100	011010 0011
D23.2	57h	111010 0101	000101 0101	D23.3	77h	111010 0011	000101 1100
D24.2	58h	110011 0101	001100 0101	D24.3	78h	110011 0011	001100 1100
D25.2	59h	100110 0101	100110 0101	D25.3	79h	100110 1100	100110 0011
D26.2	5Ah	010110 0101	010110 0101	D26.3	7Ah	010110 1100	010110 0011
D27.2	5Bh	110110 0101	001001 0101	D27.3	7Bh	110110 0011	001001 1100
D28.2	5Ch	001110 0101	001110 0101	D28.3	7Ch	001110 1100	001110 0011
D29.2	5Dh	101110 0101	010001 0101	D29.3	7Dh	101110 0011	010001 1100
D30.2	5Eh	011110 0101	100001 0101	D30.3	7Eh	011110 0011	100001 1100
D31.2	5Fh	101011 0101	010100 0101	D31.3	7Fh	101011 0011	010100 1100

(continued)

Table 18 – Valid data characters *(continued)*

Name	Byte	abcdei fghj output		Name	Byte	abcdei fghj output	
		Current rd-	Current rd+			Current rd-	Current rd+
D0.4	80h	100111 0010	011000 1101	D0.5	A0h	100111 1010	011000 1010
D1.4	81h	011101 0010	100010 1101	D1.5	A1h	011101 1010	100010 1010
D2.4	82h	101101 0010	010010 1101	D2.5	A2h	101101 1010	010010 1010
D3.4	83h	110001 1101	110001 0010	D3.5	A3h	110001 1010	110001 1010
D4.4	84h	110101 0010	001010 1101	D4.5	A4h	110101 1010	001010 1010
D5.4	85h	101001 1101	101001 0010	D5.5	A5h	101001 1010	101001 1010
D6.4	86h	011001 1101	011001 0010	D6.5	A6h	011001 1010	011001 1010
D7.4	87h	111000 1101	000111 0010	D7.5	A7h	111000 1010	000111 1010
D8.4	88h	111001 0010	000110 1101	D8.5	A8h	111001 1010	000110 1010
D9.4	89h	100101 1101	100101 0010	D9.5	A9h	100101 1010	100101 1010
D10.4	8Ah	010101 1101	010101 0010	D10.5	AAh	010101 1010	010101 1010
D11.4	8Bh	110100 1101	110100 0010	D11.5	ABh	110100 1010	110100 1010
D12.4	8Ch	001101 1101	001101 0010	D12.5	ACh	001101 1010	001101 1010
D13.4	8Dh	101100 1101	101100 0010	D13.5	ADh	101100 1010	101100 1010
D14.4	8Eh	011100 1101	011100 0010	D14.5	AEh	011100 1010	011100 1010
D15.4	8Fh	010111 0010	101000 1101	D15.5	AFh	010111 1010	101000 1010
D16.4	90h	011011 0010	100100 1101	D16.5	B0h	011011 1010	100100 1010
D17.4	91h	100011 1101	100011 0010	D17.5	B1h	100011 1010	100011 1010
D18.4	92h	010011 1101	010011 0010	D18.5	B2h	010011 1010	010011 1010
D19.4	93h	110010 1101	110010 0010	D19.5	B3h	110010 1010	110010 1010
D20.4	94h	001011 1101	001011 0010	D20.5	B4h	001011 1010	001011 1010
D21.4	95h	101010 1101	101010 0010	D21.5	B5h	101010 1010	101010 1010
D22.4	96h	011010 1101	011010 0010	D22.5	B6h	011010 1010	011010 1010
D23.4	97h	111010 0010	000101 1101	D23.5	B7h	111010 1010	000101 1010
D24.4	98h	110011 0010	001100 1101	D24.5	B8h	110011 1010	001100 1010
D25.4	99h	100110 1101	100110 0010	D25.5	B9h	100110 1010	100110 1010
D26.4	9Ah	010110 1101	010110 0010	D26.5	BAh	010110 1010	010110 1010
D27.4	9Bh	110110 0010	001001 1101	D27.5	BBh	110110 1010	001001 1010
D28.4	9Ch	001110 1101	001110 0010	D28.5	BCh	001110 1010	001110 1010
D29.4	9Dh	101110 0010	010001 1101	D29.5	BDh	101110 1010	010001 1010
D30.4	9Eh	011110 0010	100001 1101	D30.5	BEh	011110 1010	100001 1010
D31.4	9Fh	101011 0010	010100 1101	D31.5	BFh	101011 1010	010100 1010

(continued)

Table 18 – Valid data characters *(continued)*

Name	Byte	abcdei fghj output		Name	Byte	abcdei fghj output	
		Current rd-	Current rd+			Current rd-	Current rd+
D0.6	C0h	100111 0110	011000 0110	D0.7	E0h	100111 0001	011000 1110
D1.6	C1h	011101 0110	100010 0110	D1.7	E1h	011101 0001	100010 1110
D2.6	C2h	101101 0110	010010 0110	D2.7	E2h	101101 0001	010010 1110
D3.6	C3h	110001 0110	110001 0110	D3.7	E3h	110001 1110	110001 0001
D4.6	C4h	110101 0110	001010 0110	D4.7	E4h	110101 0001	001010 1110
D5.6	C5h	101001 0110	101001 0110	D5.7	E5h	101001 1110	101001 0001
D6.6	C6h	011001 0110	011001 0110	D6.7	E6h	011001 1110	011001 0001
D7.6	C7h	111000 0110	000111 0110	D7.7	E7h	111000 1110	000111 0001
D8.6	C8h	111001 0110	000110 0110	D8.7	E8h	111001 0001	000110 1110
D9.6	C9h	100101 0110	100101 0110	D9.7	E9h	100101 1110	100101 0001
D10.6	CAh	010101 0110	010101 0110	D10.7	EAh	010101 1110	010101 0001
D11.6	CBh	110100 0110	110100 0110	D11.7	EBh	110100 1110	110100 1000
D12.6	CCh	001101 0110	001101 0110	D12.7	ECh	001101 1110	001101 0001
D13.6	CDh	101100 0110	101100 0110	D13.7	EDh	101100 1110	101100 1000
D14.6	CEh	011100 0110	011100 0110	D14.7	EEh	011100 1110	011100 1000
D15.6	CFh	010111 0110	101000 0110	D15.7	EFh	010111 0001	101000 1110
D16.6	D0h	011011 0110	100100 0110	D16.7	F0h	011011 0001	100100 1110
D17.6	D1h	100011 0110	100011 0110	D17.7	F1h	100011 0111	100011 0001
D18.6	D2h	010011 0110	010011 0110	D18.7	F2h	010011 0111	010011 0001
D19.6	D3h	110010 0110	110010 0110	D19.7	F3h	110010 1110	110010 0001
D20.6	D4h	001011 0110	001011 0110	D20.7	F4h	001011 0111	001011 0001
D21.6	D5h	101010 0110	101010 0110	D21.7	F5h	101010 1110	101010 0001
D22.6	D6h	011010 0110	011010 0110	D22.7	F6h	011010 1110	011010 0001
D23.6	D7h	111010 0110	000101 0110	D23.7	F7h	111010 0001	000101 1110
D24.6	D8h	110011 0110	001100 0110	D24.7	F8h	110011 0001	001100 1110
D25.6	D9h	100110 0110	100110 0110	D25.7	F9h	100110 1110	100110 0001
D26.6	DAh	010110 0110	010110 0110	D26.7	FAh	010110 1110	010110 0001
D27.6	DBh	110110 0110	001001 0110	D27.7	FBh	110110 0001	001001 1110
D28.6	DCh	001110 0110	001110 0110	D28.7	FCh	001110 1110	001110 0001
D29.6	DDh	101110 0110	010001 0110	D29.7	FDh	101110 0001	010001 1110
D30.6	DEh	011110 0110	100001 0110	D30.7	FEh	011110 0001	100001 1110
D31.6	DFh	101011 0110	010100 0110	D31.7	FFh	101011 0001	010100 1110

(concluded)

7.2.2.5.2 Control characters

Table 19. – Valid control characters

Name	abcdei fghj output		Description
	Current rd-	Current rd+	
K28.3	001111 0011	110000 1100	Occurs only at byte 0 of all primitives except for the ALIGN primitive
K28.5	001111 1010	110000 0101	Occurs only at byte 0 of the ALIGN primitive

In Serial ATA only the K28.3 and K28.5 control characters are valid and are always used as the first byte in a four-byte primitive. The K28.3 control character is used to prefix all primitives other than the ALIGN primitive, while the K28.5 control character is used to prefix the ALIGN primitive. The encoding of characters within primitives follow the same rules as that applied to non-primitives, when calculating the running disparity between characters and between subblocks of each character within the primitive. The control characters K28.3 and K28.5 invert the current running disparity.
The running disparity at the end of the ALIGN primitive is the same as the running disparity at the beginning of the ALIGN primitive.

7.2.3 Transmission summary

7.2.3.1 Transmission order

7.2.3.1.1 Bits within a byte

The bits within an encoded character are labeled a,b,c,d,e,i,f,g,h,j. Bit “a” shall be transmitted first, followed in order by “b”, “c”, “d”, “e”, “i”, “f”, “g”, “h” and “j”. Note that bit “i” is transmitted between bits “e” and “f”, and that bit “j” is transmitted last, and not in the order that would be indicated by the letters of the alphabet. Figure 54 illustrates the transmission order within a byte

7.2.3.1.2 Bytes within a Dword

For all transmissions and receptions, Serial ATA organizes all values as Dwords. Even when representing a 32-bit value, the Dword shall be considered a set of four bytes. The transmission order of the bytes within the Dword shall be from the least-significant byte (byte 0) to the most-significant byte (byte 3). This right-to-left transmission order differs from Fibre Channel. Figure 54 illustrates how the bytes are arranged in a Dword and the order in which bits are sent.

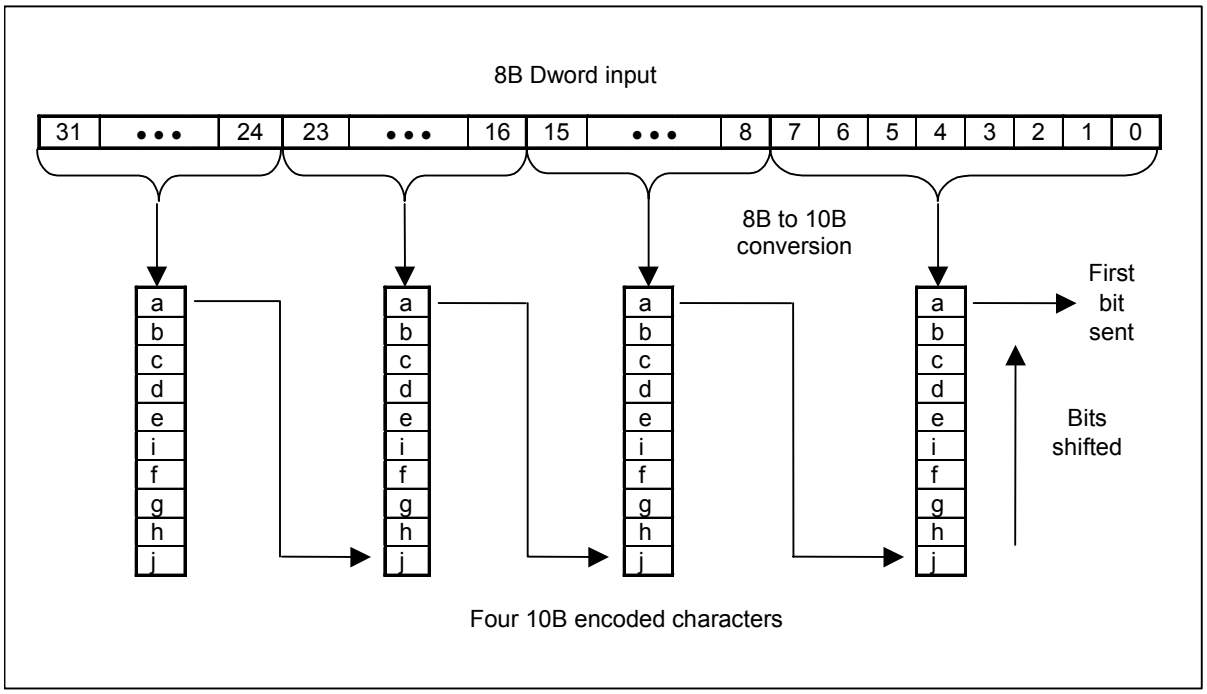


Figure 54 – Bit ordering and significance

7.2.3.1.3 Dwords within a frame

A frame (as described in Section 8.3) shall be transmitted sequentially in ascending Dword order starting with the SOF delimiter, followed by the Dwords of the frame contents, followed by the CRC, and ending with the EOF delimiter.

NOTE – While this Serial ATA standard discusses a strict hierarchy of Dword transmission as an ordered series of bytes, it is not the intent to restrict implementations from implementing a wider data path. It is possible, and even desirable, to perform transmission in word-sized fields. 8b/10b encoders with a 16(unencoded)/20(encoded) data path do exist. The only restriction is the transmission order of each byte and running disparity for each sub-block is preserved.

7.2.4 Reception summary

Upon reception of an encoded character the column corresponding to the receiver's current Running Disparity shall be searched for the encoded character value. If the encoded character value is found in the table the received encoded character shall be considered a legal character and decoded, and the decoded character value is made available to the Link layer.

If the received encoded character is not found in that column, then the encoded character shall be marked as code violation and reported to the Link layer.

7.2.4.1 Disparity and the detection of a code violation

Due to the propagation characteristics of the 8b/10b code, it is possible that although most errors will be detected, a single bit error might not be detected until several characters after the error is introduced. The following examples illustrate this effect. The first example shows a bit error being propagated two characters before being detected. The second shows a single character of propagation.

It is important to note that Serial ATA sends data in Dword increments, but the transmitter and receiver operate in units of a byte (character). The examples don't show Dword boundaries, so it is possible that an error in either of these cases could be deferred one full Dword.

The frequency of disparity errors and code violations is an indicator of channel quality and corresponds directly to the bit error rate of the physical serial link between a host and device. Implementations may elect to count such events and make them available to external firmware or software, although the method by which such counters are exposed is not defined in this specification.

Initial Running Disparity and the Running Disparity for each character is shown. In order to discover the errors note that Running Disparity is actually computed at the end of each sub-block and subsequently forwarded to the next sub-block. Footnotes indicate where the disparity error is detected. The error bit is underlined.

Figure 55 – Single bit error with two character delay

	rd	Character	rd	Character	rd	Character	rd
Transmitted character stream	-	D21.1	-	D10.2	-	D23.5	+
Transmitted bit stream	-	101010 1001	-	010101 0101	-	111010 1010	+
Received bit stream	-	101010 1011[a]	+	010101 0101	+	111010[b] 1010	+
Decoded character stream	-	D21.0	+	D10.2	+	Code violation[b]	+[c]

NOTES –
[a] Bit error introduced: 1001 → 1011
[b] Sub-blocks with non-neutral disparity must alternate polarity (i.e., + → -). In this case, rd does not alternate (it stays positive for two sub-blocks in a row). The resulting encoded character does not exist in the rd+ column in the data or control code table, and so an invalid encoded character is recognized.
[c] Running disparity must be computed on the received character regardless of the validity of the encoded character.

Figure 56 – Single bit error with one character delay

	rd	Character	rd	Character	rd	Character	rd
Transmitted character stream	-	D21.1	-	D23.4	-	D23.5	+
Transmitted bit stream	-	101010 1001	-	111010 0010	-	111010 1010	+
Received bit stream	-	101010 1011[a]	+	111010[b] 0010	-	111010 1010	+
Decoded character stream	-	D21.0	+	Code violation[b]	-	D23.5	+[c]

NOTES–
[a] Bit error introduced: 1001 => 1011
[b] Sub-blocks with non-neutral disparity must alternate polarity (i.e., + → -). In this case, rd does not alternate (it stays positive for two sub-blocks in a row). The resulting encoded character does not exist in the rd+ column in the data or control code table, and so an invalid encoded character is recognized.
[c] Running disparity must be computed on the received character regardless of the validity of the encoded character.

7.3 Transmission overview

The information on the serial line is a sequence of 8b/10b encoded characters. The smallest unit of communication is a Dword. The contents of each Dword are grouped to provide low-level control information or to transfer information between a host and an attached device.

The two types of structures are primitives and frames.

A primitive consists of a single Dword and is the simplest unit of information that may be exchanged between a host and a device. When the bytes of a primitive are encoded the resulting pattern is difficult to misinterpret as any other primitive or random pattern. Primitives are used primarily to convey real-time state information, to control the transfer of information and coordinate host / device communication. All bytes in a primitive are constants and the first byte is always a special character.

Since all of the bytes are constants, a primitive cannot be used to convey variable information. Later sections will describe the exact contents of the primitives used by Serial ATA.

A frame consists of multiple Dwords, and always starts with an SOF primitive, followed by a user payload called a Frame Information Structure (FIS), a CRC, and ends with an EOF primitive. The CRC is defined to be the last non-primitive Dword immediately preceding the EOF primitive. Some number of flow control primitives (HOLD or HOLDA, or a CONT stream to sustain a HOLD or HOLDA state) are allowed between the SOF and EOF primitives to throttle data flow for speed matching purposes.

The following figure shows an example of a sequence of transmissions.

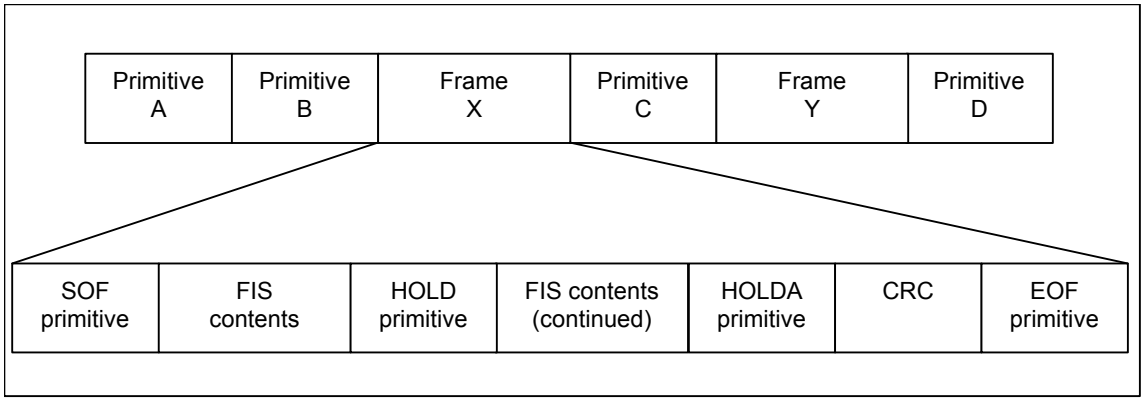


Figure 57 – Transmission structures

7.4 Primitives

7.4.1 Overview

Primitives are Dword entities that are used to control and provide status of the serial line.

Primitives always begin with a control character; all primitives use the K28.3 control character to signify the beginning of a primitive except for the ALIGN primitive which begins with the K28.5 control character. ALIGN thus represents the only primitive that contains the comma character. Following the control character, three additional characters are encoded to complete the Dword. Table 20 is a summary of the character combinations that make up each primitive.

7.4.1.1 Primitive disparity

Primitives can begin with either positive or negative disparity and end in either positive or negative disparity Normal 8b/10b encoding disparity rules are applied when encoding primitives.

The ALIGN primitive is chosen to have neutral disparity so that it can be inserted into the stream without affecting the disparity of previously encoded characters. Disparity at the end of ALIGN is the same as the ending disparity of the last character transmitted before the ALIGN primitive.

Each primitive is described and the encoding defined in the following sections.

7.4.1.2 Primitive handshakes

Some primitives are transmitted in response to receipt of other primitives to acknowledge receipt. For example, the HOLDA primitive is transmitted in response to the receipt of HOLD primitives and R_OK or R_ERR is transmitted in response to WTRM primitives. Due to the different clock domains between to two ends of the cable, the number of response primitives may not match the number of primitives to which they are responding. For example, a device may send five HOLD primitives but receive six HOLDA primitives in response. Neither the transmitter nor receiver of these primitives need count the number of primitives or match the number sent and received. There are boundary cases were a zero number of response primitives such as HOLDA may be sent.

7.4.2 Primitive descriptions

The following table contains the primitive mnemonics and a brief description of each.

Table 20. – Description of primitives

Primitive	Name	Description
ALIGN	Physical layer control	Upon receipt of an ALIGN, the physical layer readjusts internal operations as necessary to perform its functions correctly. This primitive is always sent in pairs - there is no condition where an odd number of ALIGN primitives shall be sent (except as noted for retimed loopback).
CONT	Continue repeating previous primitive	The CONT primitive allows long strings of repeated primitives to be eliminated. The CONT primitive implies that the previously received primitive be repeated as long as another primitive is not received.
DMAT	DMA terminate	This primitive is sent as a request to the transmitter to terminate a DMA data transmission early by computing a CRC on the data sent and ending with a EOF primitive. The transmitter context is assumed to remain stable after the EOF primitive has been sent.
EOF	End of frame	EOF marks the end of a frame. The previous non-primitive Dword is the CRC for the frame.
HOLD	Hold data transmission	HOLD is transmitted in place of payload data within a frame when the transmitter does not have the next payload data ready for transmission. HOLD is also transmitted on the backchannel when a receiver is not ready to receive additional payload data.
HOLDA	Hold acknowledge	This primitive is sent by a transmitter as long the HOLD primitive is received by its companion receiver.
PMACK	Power management acknowledge	Sent in response to a PMREQ_S or PMREQ_P when a receiving node is prepared to enter a power mode state.
PMNAK	Power management denial	Sent in response to a PMREQ_S or PMREQ_P when a receiving node is not prepared to enter a power mode state or when power management is not supported.
PMREQ_P	Power management request to partial	This primitive is sent continuously until PMACK or PMNAK is received. When PMACK is received, current node (host or device) will stop PMREQ_P and enters the Partial power management state.
PMREQ_S	Power management request to Slumber	This primitive is sent continuously until PMACK or PMNAK is received. When PMACK is received, current node (host or device) will stop PMREQ_S and enters the Slumber power management state.
R_ERR	Reception error	Current node (host or device) detected error in received payload.
R_IP	Reception in Progress	Current node (host or device) is receiving payload.
R_OK	Reception with no error	Current node (host or device) detected no error in received payload.

(continued)

Table 20 – Description of primitives *(concluded)*

Primitive	Name	Description
R_RDY	Receiver ready	Current node (host or device) is ready to receive payload.
SOF	Start of frame	Start of a frame. Payload and CRC follow to EOF.
SYNC	Synchronization	Synchronizing primitive – always idle.
WTRM	Wait for frame termination	After transmission of any of the EOF, the transmitter will transmit WTRM while waiting for reception status from receiver.
X_RDY	Transmission data ready	Current node (host or device) has payload ready for transmission

(concluded)

7.4.3 Primitive encoding

The following table defines the encoding for each primitive.

Table 21 – Primitive encoding

Primitive name	Byte 3 contents	Byte 2 contents	Byte 1 contents	Byte 0 contents
ALIGN	D27.3	D10.2	D10.2.	K28.5
CONT	D25.4	D25.4	D10.5	K28.3
DMAT	D22.1	D22.1	D21.5	K28.3
EOF	D21.6	D21.6	D21.5	K28.3
HOLD	D21.6	D21.6	D10.5	K28.3
HOLDA	D21.4	D21.4	D10.5	K28.3
PMACK	D21.4	D21.4	D21.4	K28.3
PMNAK	D21.7	D21.7	D21.4	K28.3
PMREQ_P	D23.0	D23.0	D21.5	K28.3
PMREQ_S	D21.3	D21.3	D21.4	K28.3
R_ERR	D22.2	D22.2	D21.5	K28.3
R_IP	D21.2	D21.2	D21.5	K28.3
R_OK	D21.1	D21.1	D21.5	K28.3
R_RDY	D10.2	D10.2	D21.4	K28.3
SOF	D23.1	D23.1	D21.5	K28.3
SYNC	D21.5	D21.5	D21.4	K28.3
WTRM	D24.2	D24.2	D21.5	K28.3
X_RDY	D23.2	D23.2	D21.5	K28.3

7.4.4 Abort primitive

Because of its unique qualities, the abort primitive - DMAT - will be discussed in detail in this section.

Parallel ATA devices can abort a DMA transfer and post an error, by negating DMARequest, updating the Error and Status registers, and possibly setting the INTRQ line. This can be performed for both reads from and writes to the device.

For example, in the case of a DMA read from device, a Serial ATA device may terminate the transfer with an EOF, and send a Register Device to Host FIS to the host, with Error and Status registers updated appropriately. In the case of a DMA write to device, the device sends a DMA Activate FIS to the host, and then after receiving an SOF, will have to accept all data until receiving an EOF from the host. Since the device cannot terminate such a transfer once started, a special abort primitive is used.

The DMA Terminate (DMAT) primitive may be sent on the back channel during transmission of a Data FIS to signal the transmitter to terminate the transfer in progress. It may be used for both host to device transfers and for device to host transfers. Reception of the DMAT signal shall cause the recipient to close the current frame by inserting the CRC and EOF, and return to the idle state.

For host to device data transfers, upon receiving the DMAT signal the host shall terminate the transfer in progress by deactivating its DMA engine and closing the frame with valid CRC and EOF. The host DMA engine shall preserve its state at the point it was deactivated so that the device may resume the transmission at a later time by transmitting another DMA Active FIS to re-activate the DMA engine. The device is responsible for either subsequently resuming the terminated transfer by transmitting another DMA Activate FIS or closing the affected command with appropriate status.

For device to host transfers, receipt of DMAT signal by the device results in permanent termination of the transfer and is not resumable. The device shall terminate the transmission in progress and close the frame with a valid CRC and EOF, and shall thereafter clean up the affected command by indicating appropriate status for that command. No facility for resuming a device to host transfer terminated with the DMAT signal is provided.

Some implementations may have an implementation-dependent latency associated with closing the affected Data FIS in response to the DMAT signal. For example, a host controller may have a small transmit FIFO, and in order for the DMA engine to accurately reflect a resumable state, the data already transferred by the DMA engine to the transmit FIFO may have to be transmitted prior to closing the affected Data FIS. Conservative designs should minimize the DMAT response latency while being tolerant of other devices having a long latency.

7.4.5 Continue primitive

In order to accommodate EMI reductions, scrambling of data is incorporated in Serial ATA as described in Section 7.5. The scrambling of data is simple, with a linear feedback shift register (LFSR) used in generating the scrambling pattern being reset at each SOF primitive, or rolling over every 2048 Dwords. However, the scrambling of primitives is not as effective or simple because of the small number of control characters available. In order to accommodate EMI reductions, repeated primitives are eliminated through the use of the CONT primitive.

Any repetitive primitive may be implied to continue repeating through the use of the CONT primitive. The recipient of the CONT primitive shall ignore all data received after the CONT primitive until the reception of any primitive, excluding ALIGN. After transmitting the CONT character, the transmitter may send any sequence of data characters to the recipient provided that no primitives are included. The reception of a CONT primitive shall cause the last valid primitive to be implied as repeated until the reception of the next valid primitive.

To improve overall protocol robustness and avoid potential timeout situations caused by a reception error in a primitive, all repeated primitives shall be transmitted a minimum of twice before a CONT primitive is transmitted. The first primitive correctly received is the initiator of any action within the receiver. This avoids scenarios, for example, where X_RDY is sent from the host, followed by a CONT, and the X_RDY is received improperly resulting in the device not returning an R_RDY and causing the system to deadlock until a timeout/reset condition occurs.

The transmission of a CONT primitive is optional, but the ability to receive and properly process the CONT primitive is required. The insertion of a single, or two repetitive primitives not followed by a CONT primitive is valid (i.e. data, data, HOLD, data).

The following primitives may be followed by a CONT: HOLD, HOLDA, PMREQ_P, PMREQ_S, R_ERR, R_IP, R_OK, R_RDY, SYNC, WTRM and X_RDY.

The host PHY initialization state machine consumes the first few received primitives before communications between the host and device have been established (see state HP7:HR_SendAlign in section 6.8.1.1.1). In order to ensure proper synchronization between the host and device after entry into the L1:L_IDLE state from the LS3:L_SendAlign state or the LPM8:L_WakeUp2 state (see section 7.6.1.2 and section 7.6.1.5), the use of the CONT primitive is not allowed after a transition from the LS3:L_SendAlign state or the LPM8:L_WakeUp2 state to the L1:L_IDLE state until either a minimum of 10 non-ALIGN primitives have been transmitted or until receipt of a primitive other than SYNC or ALIGN has been detected.

Figure 58 illustrates use of the CONT primitive in the transmission of an FIS.

Transmitter	Receiver
XXXX	XXXX
XXXX	XXXX
X_RDY	XXXX
X_RDY	XXXX
CONT	XXXX
XXXX	XXXX
XXXX	R_RDY
XXXX	R_RDY
XXXX	CONT
SOF	XXXX
TYPE	XXXX
DATA	XXXX
DATA	R_IP
DATA	R_IP
DATA	CONT
HOLD	XXXX
HOLD	XXXX
CONT	XXXX
XXXX	XXXX
XXXX	HOLDA
XXXX	HOLDA
HOLD	CONT
DATA	XXXX
DATA	XXXX
DATA	XXXX
CRC	XXXX
EOF	R_IP
WTRM	R_IP
WTRM	CONT
WTRM	XXXX
CONT	XXXX
XXXX	R_OK
XXXX	R_OK
XXXX	CONT
XXXX	XXXX
SYNC	XXXX
SYNC	XXXX
CONT	XXXX
XXXX	XXXX
XXXX	SYNC
XXXX	SYNC
XXXX	CONT
XXXX	XXXX
XXXX	XXXX

NOTE –
XXXX Scrambled data values (non-primitives)
DATA FIS payload data

Figure 58 – CONT usage example

7.4.5.1 Scrambling of data following the continue primitive

The data following the CONT shall be the output of an LFSR which implements the same polynomial as is used to scramble FIS contents. That polynomial is defined in section 7.5.

The resulting LFSR value shall be encoded using the 8b/10b rules for encoding data characters before transmission by the Link layer.

The LFSR used to supply data after the CONT primitive shall be reset to the initial value upon detection of a COMINIT or COMRESET event.

Since the data following a CONT primitive is discarded by the Link layer, the value of the LFSR is undefined between CONT primitives. That is, the LFSR result used for CONT sequence N is not required to be continuous from the last LFSR result of CONT sequence N-1.

The sequence of LFSR values used to scramble the payload contents of an FIS shall not be affected by the scrambling of data used during repeated primitive suppression. That is, the data payload LFSR shall not be advanced during repeated primitive suppression and shall only be advanced for each data payload character that is scrambled using the data payload LFSR. See section 7.5 for additional information on scrambling and repeated primitive suppression.

7.4.6 ALIGN primitive

The Link layer shall be agnostic to reception of ALIGN primitives. The Phy layer is free to consume received ALIGN primitives. Implementations where the Phy does not consume received ALIGN primitives shall effectively drop received ALIGN primitives at the input to the Link layer or shall include Link layer processing that yields behavior equivalent to the behavior produced if all received ALIGN primitives are consumed by the Phy and not presented to the Link.

7.4.7 Flow control signaling latency

There is a finite pipeline latency in a round-trip handshake across the Serial ATA interface, In order to accommodate efficient system design with sufficient buffering headroom to avoid buffer overflow in flow control situations, the maximum tolerable latency from when a receiver issues a HOLD signal until it receives the HOLDA signal from the transmitter must be specified. This allows the high-water mark to be set in the receive FIFO so as to avoid buffer overflow while avoiding excessive buffering/FIFO space.

In the case where the receiver wants to flow control the incoming data, it transmits HOLD characters on the back channel. Some number of received Dwords later, valid data ceases, and HOLDA characters are received. The larger the latency between transmitting HOLD until receiving HOLDA, the larger the receive FIFO needs to be. The maximum allowed latency from the time the MSB of the HOLD primitive is on the wire, until the MSB of the HOLDA is on the wire shall be no more than 20 Dword symbol times. The LSB is transmitted first. A receiver shall be able to accommodate reception of 20 Dwords of additional data after the time it transmits the HOLD flow control character to the transmitter, and the transmitter shall respond with a HOLDA in response to receiving a HOLD character within 20 Dword symbol times.

There is no reference design in the Serial ATA standard. The specified maximum latency figure is based on the layers and states described throughout this document. It is recognized that the Link Layer may have two separate clock domains -- transmit clock domain, and the receive clock domain. It is also recognized that a Link state machine could run at the Dword clock rate, implying synchronizers between three potential clock domains. In practice more efficient implementations would be pursued and the actual latencies may be less than indicated here. The figures represent an almost literal interpretation of the spec into logic design. A synchronizer is assumed to be a worst case of 2.99 clocks of any clock domain and is rounded to three whole clocks. The Serial ATA cable contains less than ½ Dword in generations 1 and 2, and is therefore rounded to 0. Two Dwords of

pipeline delay are assumed for the Phy, and the FIFO is assumed to run at the Link state machine rate. No synchronization is needed between the two.

The following figure outlines the origin of the 20 Dword latency specification. The example illustrates the components of a round trip delay when the receiver places a HOLD on the bus until reception of the HOLDA from the transmitter. This corresponds to the number of Dwords that the receiver must be able to accept after transmitting a HOLD character.

Receiver sends HOLD:

1 Dword	Convert to 40 bit data.
1 Dword	10b/8b conversion.
1 Dword	De scrambling.
3 Dwords	Synchronization between receive clock, and Link state machine clock.
1 Dword	Link state machine is notified that primitive has been received.
1 Dword	Link state machine takes action.
1 Dword	FIFO is notified of primitive reception.
1 Dword	FIFO stops sending data to Link layer.
1 Dword	Link is notified to insert HOLDA.
1 Dword	Link acts on notification and inserts HOLDA into data stream.
1 Dword	Scrambling.
1 Dword	8b/10b conversion.
1 Dword	Synchronize to transmit clock (3 transmit clocks, which are four times the Link state machine rate).
1 Dword	Convert to 10 bit data.
2 Dwords	Phy, transmit side.

HOLDA on the cable.

Table 22 – SRST write from host to device transmission breaking through a device to host Data FIS

Host driver	Device driver	Description
…	…	Previous activity abbreviated for clarity
R_IP	Data n	Device transmitting data
R_IP	Data n+1	
R_IP	HOLD	Device transmit FIFO empty, and flow control applied
R_IP	HOLD	Host receives and decodes HOLD flow control
HOLDA	HOLD	Host acknowledges flow control. Device internally deadlocked and no more data forthcoming (drive hung)
HOLDA	HOLD	
…	…	System in this state until host decides to reset drive
HOLDA	HOLD	Host detects SRST write to control register, needs to break deadlock
SYNC	HOLD	Host transmits SYNC primitive to abort current transmission
SYNC	HOLD	Device receives and decodes SYNC, abandons transmission in progress
SYNC	SYNC	Host sends SYNC / Device sends SYNC (both returned to idle state)
SYNC	SYNC	Host receives and decodes SYNC, can now initiate new FIS transmission
X_RDY	SYNC	Host ready to send Shadow register block registers for SRST write
X_RDY	SYNC	Device decodes X_RDY
X_RDY	R_RDY	Device indicates ready to receive
X_RDY	R_RDY	Host decodes R_RDY

SOF	R_RDY	Host starts a frame
etc	etc	etc

7.4.8 Examples

The following examples illustrate basic primitive usage. They do not show detailed lengthy sequences which will invoke the use of the CONT primitive.

Table 23. – Command Shadow Register Block register transmission example

Host driver	Device driver	Description
SYNC	SYNC	Idle condition
SYNC	SYNC	Idle condition
X_RDY	SYNC	Host ready to send Shadow register block registers
X_RDY	SYNC	Device decodes X_RDY
X_RDY	R_RDY	Device indicates ready to receive
X_RDY	R_RDY	Host decodes R_RDY
SOF	R_RDY	Host starts a frame
Hdr 0	R_RDY	Host sends Register FIS Dword 0 / device decodes SOF
Hdr 1	R_IP	Host sends Register FIS Dword 1 / device stores Hdr 0
…	…	…
Hdr n	R_IP	Host sends Register FIS Dword n / device stores Hdr n-1
CRC	R_IP	Host sends CRC / device stores Hdr n
EOF	R_IP	Host sends EOF / device stores CRC
WTRM	R_IP	Device decodes EOF
WTRM	R_IP	Device computes good CRC and releases TF contents
WTRM	R_OK	Device sends good end
WTRM	R_OK	Host decodes R_OK as good results
SYNC	R_OK	Host releases interface
SYNC	R_OK	Device decodes release by host - is allowed to release
SYNC	SYNC	Idle condition

Table 24. – Data from host to device transmission example

Host driver	Device driver	Description
SYNC	SYNC	Idle condition
SYNC	SYNC	Idle condition
X_RDY	SYNC	Host ready to send Shadow register block registers
X_RDY	SYNC	Device decodes X_RDY
X_RDY	R_RDY	Device indicates ready to receive
X_RDY	R_RDY	Host decodes R_RDY
SOF	R_RDY	Host starts a frame
Hdr 0	R_RDY	Host sends header Dword 0 / device decodes SOF
Hdr 1	R_IP	Host sends header Dword 1 / device stores header Dword 0
...	..	...
Dat x	R_IP	Host sends data Dword x / device stores data Dword (x-1)
HOLD	R_IP	Host sends HOLD / device stores data Dword (x) and decodes HOLD
HOLD	HOLDA	Device acknowledges HOLD
HOLD	HOLDA	Host decodes HOLDA – host may release HOLD at any time
Dat(n-3)	HOLDA	Host sends (n-2)th data Dword / device decodes data Dword
Dat(n-2)	R_IP	Host sends (n-1)th data Dword / device Stores (n-2)th data Dword
Dat(n-1)	R_IP	Host sends nth data Dword / device ores (n-1)th data Dword
CRC	R_IP	Host sends CRC / device stores nth data Dword
EOF	R_IP	Host sends EOF / device stores CRC
WTRM	R_IP	Device decodes EOF
WTRM	R_IP	Device computes good CRC and releases data contents
WTRM	R_OK	Device sends good end
WTRM	R_OK	Host decodes R_OK as good results
SYNC	R_OK	Host releases interface
SYNC	R_OK	Device decodes release by host - is allowed to release
SYNC	SYNC	Idle condition

Table 25. – DMA data from host to device, device terminates transmission example

Host driver	Device driver	Description
SYNC	SYNC	Idle condition
SYNC	SYNC	Idle condition
X_RDY	SYNC	Host ready to send data
X_RDY	SYNC	Device decodes X_RDY
X_RDY	R_RDY	Device indicates ready to receive
X_RDY	R_RDY	Host decodes R_RDY
SOF	R_RDY	Host starts a frame
Hdr 0	R_RDY	Host sends header Dword 0 / device decodes SOF
Dat 0	R_IP	Host sends data Dword 0
...	..	Host sends data Dword x / device stores data Dword
Dat x	DMAT	Device decides to terminate, sends DMAT
Dat x	R_IP	Host decodes DMAT – host prepares to terminate
CRC	R_IP	Host sends current CRC value
EOF	R_IP	Host sends EOF / device stores CRC
WTRM	R_IP	Device decodes EOF
WTRM	R_IP	Device computes good CRC and releases data contents
WTRM	R_OK	Device sends good end
WTRM	R_OK	Host decodes R_OK as good results
SYNC	R_OK	Host releases interface
SYNC	R_OK	Device decodes release by host - is allowed to release
SYNC	SYNC	Idle condition

7.5 CRC calculation and scrambling of FIS contents

7.5.1 CRC

The CRC (Cyclic Redundancy Check) of a frame is a Dword (32-bit) field that shall follow the last Dword of the contents of an FIS and precede the EOF primitive. The CRC calculation covers all of the FIS transport data between the SOF and EOF primitives, and excludes any intervening primitives and CONT stream contents. The CRC value shall be computed on the contents of the FIS before encoding for transmission (scrambling) and after decoding upon reception.

The CRC shall be calculated on Dword quantities. If an FIS contains an odd number of words the last word of the FIS shall be padded with zeros to a full Dword before the Dword is used in the calculation of the CRC.

The CRC shall be aligned on a Dword boundary.

The CRC shall be calculated using the following 32-bit generator polynomial:

$$G(X) = X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X + 1$$

The CRC value shall be initialized with a value of 52325032h before the calculation begins.

The maximum number of Dwords between the SOF primitive to the EOF primitive shall not exceed 2064 Dwords including the FIS type and CRC..

The contents of a frame shall be scrambled before transmission by the physical layer.

Scrambling shall be performed on Dword quantities by XORing the data to be transmitted with the output of a linear feedback shift register (LFSR). The shift register shall implement the following polynomial:

$$G(X) = X^{16} + X^{15} + X^{13} + X^{4} + 1$$

The serial shift register shall be initialized with a value of FFFFh before the first shifted output. The shift register shall be initialized to the seed value before the SOF primitive. All data words between the SOF and EOF shall be scrambled, including the CRC.

7.5.1.1 Relationship between scrambling of FIS data and repeated primitives

There are two separate scramblers used in Serial ATA. One scrambler is used for the data payload encoding and a separate scrambler is used for repeated primitive suppression. The scrambler used for data payload encoding shall maintain consistent and contiguous context over the scrambled payload characters of a frame (between SOF and EOF), and shall not have its context affected by the scrambling of data used for repeated primitive suppression.

Scrambling is applied to all data (non-primitive) Dwords. Primitives, including ALIGN, do not get scrambled and shall not advance the data payload LFSR register. Similarly, the data payload LFSR shall not be advanced during transmission of Dwords during repeated primitive suppression (i.e. after a CONT primitive). Since it is possible for a repeated primitive stream to occur in the middle of a data frame – multiple HOLD/HOLDA primitives are likely – care must be taken to ensure that the data payload LFSR is only advanced for each data payload character that it scrambles and that it is not advanced for primitives or for data characters transmitted as part of repeated primitive suppression which uses a separate scrambler.

7.5.1.2 Relationship between scrambling and CRC

The order of application of scrambling shall be as follows. For a Dword of data following the SOF primitive the Dword shall be used in the calculation of the CRC. The same Dword value shall be XORed with the scrambler output, and the resulting Dword submitted to the 8b/10b encoder for transmission. Similarly, on reception, the Dword shall be decoded using a 10b/8b decoder, the scrambler output shall be XORed with the resulting Dword, and the resulting Dword presented to the Link layer and subsequently used in calculating the CRC. The CRC Dword shall be scrambled according to the same rules.

7.6 Link layer state diagrams

7.6.1.1 Terms used in Link layer transition tables

1. LRESET: Link layer COMRESET signal
2. PhyNRdy: Phy status as defined in section 6.4.2
3. PhyRdy: Phy status as defined in section 6.4.2
4. DecErr: Bad decode of a 32 bit Dword transferred from Phy to Link
   - Invalid 10b pattern
   - Disparity error
   - Primitive with a control character in the first byte but not an allowed control character
   - Any control character in other than the first byte of the Dword
5. DatDword: A 32 bit pattern that is formed correctly, but does not have the primitive leading 10b pattern (K28.5 or K28.3).
6. COMWAKE: Signal from the out of band detector in the Phy indicating that the COMWAKE out of band signal is being detected.
7. AnyDword: A 32 bit pattern of any type - even one with DecErr received from Phy
8. The “p” subscript used in the tables designate the name immediately preceding to represent a primitive

7.6.1.2 Link idle state diagram

L1: L_IDLE[4]	Transmit $SYNC_P$	
1. Transport layer requests frame transmission and PhyRdy[2]	→	HL_SendChkRdy or DL_SendChkRdy[1]
2. Transport layer requests transition to Partial and PhyRdy[2]	→	L_TPMPartial
3. Transport layer requests transition to Slumber and PhyRdy[2]	→	L_TPMSlumber
4. X_RDY_P received from Phy	→	L_RcvWaitFifo
5. Phy layer forwards ($PMREQ\_P_P$ or $PMREQ\_S_P$) and power modes are enabled	→	L_PMOff
6. Phy layer forwards ($PMREQ\_P_P$ or $PMREQ\_S_P$) and power modes are disabled	→	L_PMDeny
7. Phy layer forwards AnyDword other than (X_RDY_P or $PMREQ\_P_P$ or $PMREQ\_S_P$) and no transmit request from Transport layer[2,3]	→	L_IDLE
8. PhyNRdy	→	L_NoCommErr

NOTE –
1. The host Link layer makes a transition to the HL_SendChkRdy state; the device Link layer makes a transition to the DL_SendChkRdy state.
2. This transition is taken even if errors such as 10b decoding errors are detected.
3. This statement also ignores any unrecognized sequences or commands not defined in this specification.
4. Upon entry to this state from the LS3:L_SendAlign state or the LPM8:L_WakeUp2 state, use of the CONT primitive is subject to restrictions as outlined in section 7.4.5.

LS1: L_NoCommErr	Post Phy not ready error to Transport layer	
1. Unconditional	→	L_NoComm

LS2: L_NoComm	Transmit $ALIGN_P$[1]	
1. PhyNRdy	→	L_NoComm
2. PhyRdy	→	L_SendAlign
NOTE – 1. Also deactivate any signal for Phy layer to abort operation		

LS3: L_SendAlign	Transmit $ALIGN_P$	
1. PhyNRdy	→	L_NoCommErr
2. PhyRdy	→	L_IDLE

LS4: L_RESET	Reset Link state to initial conditions	
1. LRESET Link reset signal asserted	→	L_RESET
2. LRESET Link reset signal deasserted	→	L_NoComm
NOTE – 1. This state is entered ANYTIME the link reset control is active.		

L1: L_IDLE state: This state is entered when a frame transmission has been completed by the Link layer.

When in this state, the Link layer transmits the SYNC primitive and waits for an X_RDY primitive from the Phy layer or a frame transmission request from the Transport layer.

Transition L1:1a: When the host Link layer receives a request to transmit a frame from the Transport layer and the Phy layer is ready, the Link layer shall make a transition to the LT1: HL_SendChkRdy state.

Transition L1:1b: When the device Link layer receives a request to transmit a frame from the Transport layer and the Phy layer is ready, the Link layer shall make a transition to the LT2: DL_SendChkRdy state.

Transition L1:2: When the Link layer receives a request to enter the Partial power mode from the Transport layer and the Phy layer is ready, the Link layer shall make a transition to the L_TPMPartial state.

Transition L1:3: When the Link layer receives a request to enter the Slumber power mode from the Transport layer and the Phy layer is ready, the Link layer shall make a transition to the L_TPMSlumber state.

Transition L1:4: When the Link layer receives an X_RDY from the Phy layer, the Link layer shall make a transition to the LR2: L_RcvWaitFifo state.

Transition L1:5: When the Link layer receives a PMREQ_P or PMREQ_S from the Phy layer and is enabled to perform power management modes, the Link layer shall make a transition to the LPM3: L_PMOff state.

Transition L1:6: When the Link layer receives a PMREQ_P or a PMREQ_S from the Phy layer and is not enabled to perform power management modes, the Link layer shall make a transition to the LR0: L_PMDeny state.

Transition L1:7: When the Link layer does not receive a request to transmit a frame from the Transport layer, does not receive a request to go to a power mode from the Transport layer, does not receive an X_RDY from the Phy layer or does not receive a PMREQ_x from the Phy layer the Link layer shall make a transition to the L1: L_IDLE state.

Transition L1:8: If the Phy layer becomes not ready even if the Transport layer is requesting an operation, the Link layer transitions to the L_NoCommErr state.

LS1: L_NoCommErr state: This state is entered upon detection of a non ready condition of the Phy layer while attempting to process another state. The entry into this state heralds a relatively serious error condition in the Link layer. This state is executed only once so as to pass on the error condition up to the Transport layer.

Transition LS1:1: The transition is made to LS1:L_NoComm unconditionally.

LS2: L_NoComm state: This state is entered directly from the LS1:L_NoCommErr state or the LS4:L_RESET State. The Link Layer remains in this state until the Phy signals that it has established communications and is ready.

Transition LS2:1: For as long as the Phy layer stays not ready, the transition is made to LS2: L_NoComm.

Transition LS2:2: When the Phy layer signals it is ready, a transition is made to LS3: L_SendAlign.

LS3: L_SendAlign state: This state is entered whenever an ALIGN needs to be sent to the Phy layer.

Transition LS3:1: If the Phy layer becomes not ready, then a transition is made to LS1: L_NoCommErr.

Transition LS3:2: If the Phy indicates that it is ready, a transition is made to the L1: L_IDLE state.

LS4: L_RESET state: This state is entered whenever the Link LRESET control is active. All Link layer hardware is initialized to and held at a known state/value. While in this state all requests or triggers from other layers are ignored. While in this state, the Phy reset signal is also asserted.

Transition LS4:1: While the RESET control is active a transition is made back to the LS4: L_RESET state.

Transition LS4:2: When the RESET control goes inactive a transition is made to the LS2: L_NoComm state.

7.6.1.3 Link transmit state diagram

LT1: HL_SendChkRdy	Transmit X_RDY_P	
1. R_RDY_P received from Phy	→	L_SendSOF
2. X_RDY_P received from Phy	→	L_RcvWaitFifo
3. AnyDword other than (R_RDY_P or X_RDY_P)[1] received from Phy layer	→	HL_SendChkRdy
4. PhyNRdy	→	L_NoCommErr[2]
1. NOTE –Any received errors such as 10b decoding errors and invalid primitives are ignored 2. The Link layer shall notify the Transport layer of the condition and fail the attempted transfer		

LT2: DL_SendChkRdy	Transmit X_RDY_P	
1. R_RDY_P received from Phy	→	L_SendSOF
2. AnyDword other than R_RDY_P received from Phy	→	DL_SendChkRdy
3. PhyNRdy	→	L_NoCommErr

LT3: L_SendSOF	Transmit SOF_P	
1. PhyRdy[1]	→	L_SendData
2. PhyNRdy	→	L_NoCommErr[2]
3. $SYNC_P$ received from Phy	→	L_IDLE[2]
NOTE – 1. Any received errors such as 10b decoding errors and invalid primitives are ignored 2. The Link layer shall notify the Transport layer of the condition and fail the attempted transfer		

LT4: L_SendData	Transmit data $Dword_W$	
1. More data to transmit and AnyDword other than ($HOLD_P$ or $DMAT_p$ or $SYNC_P$) received from Phy[1,2]	→	L_SendData
2. More data to transmit and $HOLD_P$ received from Phy	→	L_RcvrHold
3. Data transmit not complete and data not ready to transmit and AnyDword other than $SYNC_P$ received from Phy	→	L_SendHold
4. $DMAT_p$ received from Phy or data transmit complete and AnyDword other than $SYNC_P$ received from Phy	→	L_SendCRC
5. $SYNC_P$ received from Phy	→	L_IDLE[3]
6. PhyNRdy	→	L_NoCommErr[3]
7. Host Transport Layer indicates request to transmit Control Register frame[4]	→	L_IDLE

NOTE –
1. Any received errors such as 10b decoding errors and invalid primitives are ignored
2. This makes possible a back channel during this time
3. The Link layer shall notify the Transport layer of the condition and fail the attempted transfer.
4. When this condition is true, the associated transition has priority over all other transitions exiting this state.

LT5: L_RcvrHold	Transmit $HOLDA_P$	
1. More data to transmit and AnyDword other than ($HOLD_p$ or $SYNC_p$ or $DMAT_p$) received from Phy	→	L_SendData
2. More data to transmit and $HOLD_P$ received from Phy or DecErr	→	L_RcvrHold
3. More data to transmit and $SYNC_P$ received from Phy	→	L_IDLE[1]
4. More data to transmit and $DMAT_p$ received from Phy	→	L_SendCRC
5. PhyNRdy	→	L_NoCommErr[1]
6. Host Transport Layer indicates request to transmit Control Register frame[2]	→	L_IDLE
7. $SYNC_P$ received from Phy	→	L_IDLE[1]

NOTE –
1. The Link layer shall notify the Transport layer of the condition and fail the attempted transfer
2. When this condition is true, the associated transition has priority over all other transitions exiting this state.

LT6: L_SendHold	Transmit $HOLD_P$	
1. More data ready to transmit and AnyDword other than ($HOLD_P$ or $SYNC_P$) received from Phy	→	L_SendData
2. More data ready to transmit and $HOLD_P$ received from Phy	→	L_RcvrHold
3. Data transmit not complete and data not ready to transmit and AnyDword other than ($SYNC_P$ or $DMAT_p$) received from Phy	→	L_SendHold
4. DMATp received from Phy	→	L_SendCRC
5. $SYNC_P$ received from Phy	→	L_IDLE[1]
6. PhyNRdy	→	L_NoCommErr[1]
7. Host Transport Layer indicates request to transmit Control Register frame[2]	→	L_IDLE

NOTE –
1. The Link layer shall notify the Transport layer of the condition and fail the attempted transfer.
2. When this condition is true, the associated transition has priority over all other transitions exiting this state.

LT7: L_SendCRC	Transmit CRC_W	
1. PhyRdy and $SYNC_P$ not received from Phy	→	L_SendEOF
2. PhyNRdy	→	L_NoCommErr[1]
3. PhyRdy and $SYNC_P$ received from Phy	→	L_IDLE[1]

NOTE –
1. The Link layer shall notify the Transport layer of the condition and fail the attempted transfer.

LT8: L_SendEOF	Transmit EOF_P	
1. PhyRdy and $SYNC_P$ not received from Phy	→	L_Wait
2. PhyNRdy	→	L_NoCommErr[1]
3. PhyRdy and $SYNC_P$ received from Phy	→	L_IDLE[1]

NOTE –
1. The Link layer shall notify the Transport layer of the condition and fail the attempted transfer.

LT9: L_Wait	Transmit $WTRM_P$	
1. R_OK_P received from Phy	→	L_IDLE (good status)
2. R_ERR_P received from Phy	→	L_IDLE (bad status)
3. $SYNC_P$ received from Phy	→	L_IDLE[1]
4. AnyDword other than (R_OK_P or R_ERR_P , or $SYNC_P$) received from Phy	→	L_Wait
5. PhyNRdy	→	L_NoCommErr[1]

NOTE –
1. The Link layer shall notify the Transport layer of the condition and fail the attempted transfer.

LT1: HL_SendChkRdy state: This state is entered when a frame transmission has been requested by the host Transport layer.

When in this state, the Link layer transmits an X_RDY primitive and waits for an X_RDY primitive or R_RDY primitive from the Phy layer.

> NOTE – It is possible that both the host and the device simultaneously request frame transmission by transmitting X_RDY. If the host receives X_RDY while transmitting X_RDY, the host shall back off and enter the L_RcvChkRdy state, postponing its desired frame transmission until the device has completed its frame transmission and the bus is idle.

Transition LT1:1: When the host Link layer receives an R_RDY primitive from the Phy layer, the Link layer shall make a transition to the LT3: L_SendSOF state.

Transition LT1:2: When the host Link layer receives an X_RDY primitive from the Phy layer, the Link layer shall make a transition to the LR2: L_RcvWaitFifo state.

Transition LT1:3: When the host Link layer receives any Dword other than an R_RDY or an X_RDY primitive from the Phy layer, the Link layer shall make a transition to the LT1: HL_SendChkRdy state.

Transition LT1:4: When the host Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

LT2: DL_SendChkRdy state: This state is entered when a frame transmission has been requested by the device Transport layer.

When in this state, the Link layer transmits an X_RDY primitive and waits for an R_RDY primitive from the Phy layer.

Transition LT2:1: When the device Link layer receives an R_RDY primitive from the Phy layer, the Link layer shall make a transition to the LT3: L_SendSOF state.

Transition LT2:2: When the device Link layer does not receive an R_RDY primitive from the Phy layer, the Link layer shall make a transition to the LT2: DL_SendChkRdy state.

Transition LT2:3: When the device Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

LT3: L_SendSOF state: This state is entered an R_RDY primitive has been received from the Phy layer.

When in this state, the Link layer transmits an SOF primitive.

Transition LT3:1: When the device Link layer has transmitted an SOF primitive, the Link layer shall make a transition to the LT4: L_SendDATA state.

Transition LT3:2: When the Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

Transition LT3:3: When the Link layer receives a SYNC primitive from the Phy layer, the Link layer shall notify the Transport layer of the illegal transition error condition and shall make a transition to the L1:L_IDLE state.

LT4: L_SendData state: This state is entered when an SOF primitive has been transmitted.

When in this state, the Link layer takes a data Dword from the Transport layer, encodes the Dword, and transmits it. The Dword is also entered into the CRC calculation before encoding.

Transition LT4:1: When the Link layer receives any Dword other than a HOLD, DMAT, or SYNC primitive from the Phy layer and the Transport layer indicates a Dword is available for transfer, the Link layer shall make a transition to the LT4: L_SendData state. The DMAT signal is advisory and data transmission should be halted at the earliest opportunity but is not required to cease immediately. It is therefore allowable to stay in the LT4: L_SendData state when there is more data to transmit and DMAT is received.

Transition LT4:2: When the device Link layer receives a HOLD primitive from the Phy layer, the Link layer shall make a transition to the LT5: L_RcvrHold state.

Transition LT4:3: When the Transport layer indicates that the next Dword is not available to transfer and any Dword other than a SYNC primitive has been received from the Phy layer, the Link layer shall make a transition to the LT6: L_SendHold state.

Transition LT4:4: When the Transport layer indicates that all data for the frame has been transferred and any Dword other than a SYNC primitive has been received from the Phy layer, the Link layer shall make a transition to the LT7: L_SendCRC state. When the Link layer receives a DMAT primitive from the Phy layer, it shall notify the Transport layer and terminate the transmission in progress as described in section 7.4.4 and shall transition to the LT7: L_SendCRC state.

Transition LT4:5: When the Link layer receives a SYNC primitive from the Phy layer, the Link layer shall notify the Transport layer of the illegal transition error condition and shall make a transition to the L1:L_IDLE state
Transition LT4:6: When the Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

Transition LT4:7: When the host Link layer receives notification from the host Transport layer that a Control Register FIS is pending for transmission (typically due to SRST being toggled), the current transfer shall be aborted and a transition to the L_IDLE state shall be made. The return to the L_IDLE state results in the transmission of SYNC primitives, which causes the receiving Link layer to also transition to the L_IDLE state. The Control Register FIS may then be transmitted as a Register FIS Host to Device.

LT5: L_RcvrHold state: This state is entered when a HOLD primitive has been received from the Phy layer.

When in this state, the Link layer shall transmit the HOLDA primitive.

Transition LT5:1: When the Link layer receives any Dword other than a HOLD, SYNC, or a DMAT primitive from the Phy layer and the Transport layer indicates that a Dword is available for transfer, the Link layer shall make a transition to the LT4: L_SendData state.

Transition LT5:2: When the device Link layer receives a HOLD primitive from the Phy layer or a decoding error was detected, the Link layer shall make a transition to the LT5: L_RcvrHold state.

Transition LT5:3: When the Link layer receives a SYNC primitive from the Phy layer, the Link layer shall make a transition to the L1: L_IDLE state. The Transport layer shall be notified of the illegal transition error condition.

Transition LT5:4: When the Link layer receives a DMAT primitive from the Phy layer, it shall notify the Transport layer and terminate the transmission in progress as described in section 7.4.4 and shall transition to the LT7: L_SendCRC state.

Transition LT5:5: When the Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

Transition LT5:6: When the host Link layer receives notification from the host Transport layer that a Control Register FIS is pending for transmission (typically due to SRST being toggled), the current transfer shall be aborted and a transition to the L_IDLE state shall be made. The return to the L_IDLE state results in the transmission of SYNC primitives, which causes the receiving Link layer to also transition to the L_IDLE state. The Control Register FIS may then be transmitted as a Register FIS Host to Device.

Transition LT5:7: When the Link layer receives a SYNC primitive from the Phy layer, the Link layer shall notify the Transport layer of the illegal transition error condition and shall make a transition to the L1:L_IDLE state.

LT6: L_SendHold state: This state is entered when the Transport layer indicates a Dword is not available for transfer and a HOLD primitive has not been received from the Phy layer.

When in this state, the Link layer shall transmit the HOLD primitive.

Transition LT6:1: When the Link layer receives any Dword other than a HOLD or SYNC primitive from the Phy layer and the Transport layer indicates that a Dword is available for transfer, the Link layer shall make a transition to the LT4: L_SendData state.

Transition LT6:2: When the Link layer receives a HOLD primitive from the Phy layer and the Transport layer indicates a Dword is available for transfer, the Link layer shall make a transition to the LT5: L_RcvrHold state.

Transition LT6:3: When the Transport layer indicates that a Dword is not available for transfer and any Dword other than a SYNC primitive is received from the Phy layer, the Link layer shall make a transition to the LT6: L_SendHold state.

Transition LT6:4: When the Link layer receives a DMAT primitive from the Phy layer, it shall notify the Transport layer and terminate the transmission in progress as described in section 7.4.4 and shall transition to the LT7:L_SendCRC state.

Transition LT6:5: When the Link layer receives a SYNC primitive from the Phy layer, the Link layer shall make a transition to the L1: L_IDLE state. The Transport layer shall be notified of the illegal transition error condition.

Transition LT6:6: When the Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

Transition LT6:7: When the host Link layer receives notification from the host Transport layer that a Control Register FIS is pending for transmission (typically due to SRST being toggled), the current transfer shall be aborted and a transition to the L_IDLE state shall be made. The return to the L_IDLE state results in the transmission of SYNC primitives, which causes the receiving Link layer to also transition to the L_IDLE state. The Control Register FIS may then be transmitted as a Register FIS Host to Device.

LT7: L_SendCRC state: This state is entered when the Transport layer indicates that all data Dwords have been transferred for this frame.

When in this state, the Link layer shall transmit the calculated CRC for the frame.

Transition LT7:1: When the CRC has been transmitted, the Link layer shall make a transition to the LT8: L_SendEOF state.

Transition LT7:2: When the Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

Transition LT7:3: When the Link layer receives a SYNC primitive from the Phy layer, the Link layer shall notify the Transport layer of the illegal transition error condition and shall make a transition to the L1:L_IDLE state.

LT8: L_SendEOF state: This state is entered when the CRC for the frame has been transmitted.

When in this state, the Link layer shall transmit the EOF primitive.

Transition LT8:1: When the EOF primitive has been transmitted, the Link layer shall make a transition to the LT9: L_Wait state.

Transition LT8:2: When the Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

Transition LT8:3: When the Link layer receives a SYNC primitive from the Phy layer, the Link layer shall notify the Transport layer of the illegal transition error condition and shall make a transition to the L1:L_IDLE state.

LT9: L_Wait state: This state is entered when the EOF primitive has been transmitted.

When in this state, the Link layer shall transmit the WTRM primitive.

Transition LT9:1: When the Link layer receives a R_OK primitive from the Phy layer, the Link layer shall notify the Transport layer and make a transition to the L1: L_IDLE state.

Transition LT9:2: When the Link layer receives a R_ERR primitive from the Phy layer, the Link layer shall notify the Transport layer and make a transition to the L1: L_IDLE state.

Transition LT9:3: When the Link layer receives a SYNC primitive from the Phy layer, the Link layer shall notify the Transport layer and make a transition to the L1: L_IDLE state.

Transition LT9:4: When the Link layer receives any Dword other than an R_OK, R_ERR, or SYNC primitive from the Phy layer, the Link layer shall make a transition to the LT9: L_Wait state.

Transition LT9:5: When the Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

7.6.1.4 Link receive state diagram

LR1: L_RcvChkRdy	Transmit R_RDY_P	
1. X_RDY_P received from Phy	→	L_RcvChkRdy
2. SOF_P received from Phy	→	L_RcvData
3. Any Dword other than (X_RDY_P or SOF_P) received from Phy	→	L_IDLE
4. PhyNRdy	→	L_NoCommErr[1]
NOTE – 1. The Link layer shall notify the Transport layer of the condition and fail the attempted transfer.		

LR2: L_ RcvWaitFifo	Transmit $SYNC_P$	
1. X_RDY_P received from Phy and Fifo space available	→	L_RcvChkRdy
2. X_RDY_P received from Phy and Fifo space not available	→	L_RcvWaitFifo
3. Any Dword other than X_RDY_P received from Phy	→	L_IDLE
4. PhyNRdy	→	L_NoCommErr[1]
NOTE – 1. The Link layer shall notify the Transport layer of the condition and fail the attempted transfer.		

LR3: L_RcvData	Transmit R_IP_P or $DMAT_p$[1]	
1. ($DatDword_W$ received from Phy and FIFO space) or $HOLDA_P$ received from Phy	→	L_RcvData
2. $DatDword_W$ received from Phy and insufficient FIFO space	→	L_Hold
3. $HOLD_P$ received from Phy	→	L_SendHold
4. EOF_P received from Phy	→	L_RcvEOF
5. $WTRM_P$ received from Phy	→	L_BadEnd
6. $SYNC_P$ received from Phy	→	L_IDLE
7. AnyDword other than ($HOLD_P$ or EOF_P or $HOLDA_P$ or $SYNC_P$ or $WTRM_P$) received from Phy	→	L_RcvData
8. PhyNRdy	→	L_NoCommErr[2]
9. Host Transport Layer indicates request to transmit Control Register frame	→	L_IDLE
NOTE – 1. If the Transport layer signals that it wishes to terminate the transfer, DMAT is transmitted in place of R_IP. 2. The Link layer shall notify the Transport layer of the condition and fail the attempted transfer.		

LR4: L_Hold	Transmit $HOLD_P$	
1. FIFO space available and AnyDword other than $HOLD_P$ or EOF_p received from Phy	→	L_RcvData
2. FIFO space available and $HOLD_P$ received from Phy	→	L_SendHold
3. EOF_p received from Phy	→	L_RcvEOF
4. No FIFO space available and EOF_p not received from Phy and $SYNC_P$ not received from Phy and PhyRdy	→	L_Hold
5. PhyNRdy	→	L_NoCommErr[1]
6. $SYNC_P$ received from Phy	→	L_IDLE
7. Host Transport Layer indicates request to transmit Control Register frame	→	L_IDLE
NOTE – 1. The Link layer shall notify the Transport layer of the condition and fail the attempted transfer.		

LR5: L_SendHold	Transmit $HOLDA_P$ or $DMAT_p$[1]	
1. AnyDword other than ($HOLD_P$ or EOF_P or $SYNC_P$) received from Phy	→	L_RcvData
2. $HOLD_P$ received from Phy	→	L_SendHold
3. EOF_P received from Phy	→	L_RcvEOF
4. $SYNC_P$ received from Phy	→	L_IDLE
5. PhyNRdy	→	L_NoCommErr[2]
6. Host Transport Layer indicates request to transmit Control Register frame	→	L_IDLE
NOTE – 1. If the Transport layer signals that it wishes to terminate the transfer, DMAT is transmitted in place of HOLDA. 2. The Link layer shall notify the Transport layer of the condition and fail the attempted transfer.		

LR6: L_RcvEOF	Transmit R_IP_P	
1. CRC check not complete	→	L_RcvEOF
2. CRC good	→	L_GoodCRC
3. CRC bad	→	L_BadEnd
4. PhyNRdy	→	L_NoCommErr

LR7: L_GoodCRC	Transmit R_IP_P	
1. Transport Layer indicated good result	→	L_GoodEnd
2. Transport Layer indicates unrecognized FIS	→	L_BadEnd
3. Transport Layer has yet to respond	→	L_GoodCRC
4. PhyNRdy	→	L_NoCommErr
5. Transport or Link Layer indicated error detected during reception of recognized FIS	→	L_BadEnd
6. $SYNC_P$ received from Phy	→	L_IDLE
NOTE 1. Upon entering this state for the first time, the Link Layer shall notify the Transport Layer that the CRC for this frame is valid.		

LR8: L_GoodEnd	Transmit R_OK_P	
1. $SYNC_P$ received from Phy	→	L_IDLE
2. AnyDword other than $SYNC_P$ received from Phy	→	L_GoodEnd
3. PhyNRdy	→	L_NoCommErr

LR9: L_BadEnd	Transmit R_ERR_P	
1. $SYNC_P$ received from Phy	→	L_IDLE
2. AnyDword other than $SYNC_P$ received from Phy	→	L_BadEnd
3. PhyNRdy	→	L_NoCommErr

LR1: L_RcvChkRdy state: This state is entered when an X_RDY primitive has been received from the Phy layer.

When in this state, the Link layer shall transmit an R_RDY primitive and wait for an SOF primitive from the Phy layer.

Transition LR1:1: When the Link layer receives an X_RDY primitive from the Phy layer, the Link layer shall make a transition to the LR1: L_RcvChkRdy state.

Transition LR1:2: When the Link layer receives an SOF primitive from the Phy layer, the Link layer shall make a transition to the LR2: L_RcvData state.

Transition LR1:3: When the Link layer receives any Dword other than an X_RDY or SOF primitive from the Phy layer, the Link layer shall notify the Transport layer of the condition and make a transition to the L1: L_IDLE state.

Transition LR1:4: When the Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

LR2: L_RcvWaitFifo state: This state is entered when an X_RDY has been received, and the FIFO is not ready to receive an FIS.

When in this state, the Link layer shall transmit the SYNC primitive.

Transition LR2:1: When the Link layer receives a X_RDY primitive from the Phy layer and the FIFO is ready to accept data, the Link layer shall make a transition to the LR1: L_RcvChkRdy state.

Transition LR2:2: When the Link layer receives a X_RDY primitive from the Phy layer and the FIFO is not ready to accept data, the Link layer shall make a transition to the LR2: L_RcvWaitFifo state.

Transition LR2:3: When the Link layer receives any Dword other than an X_RDY primitive from the Phy layer, the Link layer shall notify the Transport layer of the condition and make a transition to the L1: L_IDLE state.

Transition LR2:4: When the Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

LR3: L_RcvData state: This state is entered when an SOF primitive has been received from the Phy layer.

When in this state, the Link layer receives an encoded character sequence from the Phy layer, decodes it into a Dword, and passes the Dword to the Transport layer. The Dword is also entered into the CRC calculation. When in this state the Link layer either transmits a R_IP primitive to signal transmission to continue or transmits a DMAT primitive to signal the transmitter to terminate the transmission.

Transition LR3:1: When the Transport layer indicates that space is available in its FIFO, the Link layer shall make a transition to the LR3: L_RcvData state.

Transition LR3:2: When the Transport layer indicates that sufficient space is not available in its FIFO, the Link layer shall make a transition to the LR4: L_Hold state.

Transition LR3:3: When the Link layer receives a HOLD primitive from the Phy layer, the Link layer shall make a transition to the LR5: L_SendHold state.

Transition LR3:4: When the Link layer receives an EOF primitive from the Phy layer, the Link layer shall make a transition to the LR6: L_RcvEOF state.

Transition LR3:5: When the Link layer receives a WTRM primitive from the Phy layer, the Link layer shall make a transition to the LR8: L_BadEnd state.

Transition LR3:6: When the Link layer receives a SYNC primitive from the Phy layer, the Link layer shall notify the Transport layer that reception was aborted and shall make a transition to the L1: L_IDLE state.

Transition LR3:7: When the Link layer receives any Dword other than a HOLD, HOLDA, EOF, or SYNC primitive from the Phy layer, the Link layer shall make a transition to the LR3: L_RcvData state.

Transition LR3:8: When the Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

Transition LR3:9: When the host Link layer receives notification from the host Transport layer that a Control Register Frame is pending for transmission (typically due to SRST being toggled), a transition to the L_IDLE state shall be made. The return to the L_IDLE state results in the transmission of SYNC primitives, which causes the transmitting Link layer to also transition to the L_IDLE state. The Control Register FIS may then be transmitted as a Register FIS from host to device.

LR4: L_Hold state: This state is entered when the Transport layer indicates that sufficient space is not available in its receive FIFO.

When in this state, the Link layer shall transmit the HOLD primitive and may receive an encoded character from the Phy layer.

Transition LR4:1: When the Link layer receives any Dword other than a HOLD primitive from the Phy layer and the Transport layer indicates that sufficient space is now available in its receive FIFO, the Link layer shall make a transition to the LR3: L_RcvData state.

Transition LR4:2: When the Link layer receives a HOLD primitive from the Phy layer and the Transport layer indicates that space is now available in its FIFO, the Link layer shall make a transition to the LR5: L_SendHold state.

Transition LR4:3: When the Link layer receives a EOF primitive from the Phy layer, the Link layer shall make a transition to the LR6: L_RcvEOF state. Note that due to pipeline latency, an EOF may be received when in the L_Hold state in which case the receiving Link shall use its FIFO headroom to receive the EOF and close the frame reception.

Transition LR4:4: When the Transport layer indicates that there is not sufficient space available in its FIFO and the Phy layer is ready, the Link layer shall make a transition to the LR4: L_Hold state.

Transition LR4:5: When the Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

Transition LR4:6: When the Link layer receives a SYNC primitive from the Phy layer, the Link layer shall notify the Transport layer of the illegal transition error condition and shall make a transition to the L1: L_IDLE state.

Transition LR4:7: When the host Link layer receives notification from the host Transport layer that a Control Register Frame is pending for transmission (typically due to SRST being toggled), a transition to the L_IDLE state shall be made. The return to the L_IDLE state results in the transmission of SYNC primitives, which causes the transmitting Link layer to also transition to the L_IDLE state. The Control Register FIS may then be transmitted as a Register FIS from host to device.

LR5: L_SendHold state: This state is entered when a HOLD primitive has been received from the Phy layer.

When in this state, the Link layer shall either transmit the HOLDA primitive to signal transmission to proceed when the transmitter becomes ready or transmit a DMAT primitive to signal the transmitter to terminate the transmission.

Transition LR5:1: When the Link layer receives any Dword other than a HOLD or SYNC primitive from the Phy layer, the Link layer shall make a transition to the LR3: L_RcvData state.

Transition LR5:2: When the Link layer receives a HOLD primitive from the Phy layer, the Link layer shall make a transition to the LR5: L_SendHold state.

Transition LR5:3: When the Link layer receives a EOF primitive from the Phy layer, the Link layer shall make a transition to the LR6: L_RcvEOF state.

Transition LR5:4: When the Link layer receives a SYNC primitive from the Phy layer, the Link layer shall make a transition to the L1: L_IDLE state. The Transport layer shall be notified of the illegal transition error condition.

Transition LR5:5: When the Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

Transition LR5:6: When the host Link layer receives notification from the host Transport layer that a Control Register Frame is pending for transmission (typically due to SRST being toggled), a transition to the L_IDLE state shall be made. The return to the L_IDLE state results in the transmission of SYNC primitives, which causes the transmitting Link layer to also transition to the L_IDLE state. The Control Register FIS may then be transmitted as a Register FIS from host to device.

LR6: L_RcvEOF state: This state is entered when the Link layer has received an EOF primitive from the Phy layer.

When in this state, the Link layer shall check the calculated CRC for the frame and transmit one or more R_IP primitives.

Transition LR6:1: If the CRC calculation and check is not yet completed, the Link layer shall make a transition to the LR6: L_RcvEOF state.

Transition LR6:2: When the CRC indicates no error, the Link layer shall notify the Transport layer and make a transition to the LR7: L_GoodCRC state.

Transition LR6:3: When the CRC indicates an error has occurred, the Link layer shall notify the Transport layer and make a transition to the LR8: L_BadEnd state.

Transition LR6:4: When the Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

LR7: L_GoodCRC state: This state is entered when the CRC for the frame has been checked and determined to be good.

When in this state, the Link layer shall wait for the Transport Layer to check the frame and transmit one or more R_IP primitives.

Transition LR7:1: When the Transport Layer indicates a good result, the Link Layer shall transition to the LR8: L_GoodEnd state.

Transition LR7:2: When the Transport Layer indicates an unrecognized FIS, the Link Layer shall transition to the LR9: L_BadEnd state.

Transition LR7:3: If the Transport Layer has not supplied status, then the Link Layer shall transition to the LR7: L_GoodCRC state.

Transition LR7:4: When the Link layer detects that the Phy layer is not ready, the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

Transition LR7:5: When the Transport layer or Link layer indicates an error was encountered during the reception of the recognized FIS, the Link layer shall transition to the LR9: L_BadEnd state.

Transition LR7:6: When the Link layer receives a SYNC primitive from the Phy layer, the Link layer shall notify the Transport layer of the illegal transition error condition and shall make a transition to the L1: L_IDLE state.

LR8: L_GoodEnd state: This state is entered when the CRC for the frame has been checked and determined to be good.

When in this state, the Link layer shall transmit the R_OK primitive.

Transition LR8:1: When the Link layer receives a SYNC primitive from the Phy layer, the Link layer shall make a transition to the L1: L_IDLE state.

Transition LR8:2: When the Link layer receives any Dword other than a SYNC primitive from the Phy layer, the Link layer shall make a transition to the LR7: L_GoodEnd state.

Transition LR8:3: When the Link layer detects that the Phy layer is not ready, the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

LR9: L_BadEnd state: This state is entered when the CRC for the frame has been checked and determined to be bad or when the Transport layer has notified the Link layer that the received FIS is invalid.

When in this state, the Link layer shall transmit the R_ERR primitive.

Transition LR9:1: When the Link layer receives a SYNC primitive from the Phy layer, the Link layer shall make a transition to the L1: L_IDLE state.

Transition LR9:2: When the Link layer receives any Dword other than a SYNC primitive from the Phy layer, the Link layer shall make a transition to the LR8:BadEnd state.

Transition LR9:3: When the Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

7.6.1.5 Link power mode state diagram

LPM1: L_TPMPartial	Transmit $PMREQ\_P_P$	
1. $PMACK_P$ received from Phy	→	L_ChkPhyRdy
2. X_RDY_P received from Phy	→	L_RcvWaitFifo[1]
3. $SYNC_P$ or R_OK_P received from Phy	→	L_TPMPartial
4. AnyDword other than ($PMACK_P$ or X_RDY_P or $SYNC_P$ or R_OK_P or $PMREQ\_P_p$[3] or $PMREQ\_S_p$[3])[1] received from Phy layer	→	L_IDLE
5. $PMREQ\_P_p$ or $PMREQ\_S_p$ received from Phy	→	L_TPMPartial[3]
6. PhyNRdy	→	L_NoCommErr[2]

NOTE –
1. This transition aborts the request from the Transport layer to enter a power mode. A status indication to the Transport layer of this event is required
2. This is an unexpected transition and constitutes an error condition. An error condition needs to be sent to the Transport layer as a result.
3. If $PMREQ\_P_p$ or $PMREQ\_S_p$ is received, the host shall make a transition to the L_IDLE state, but the device shall make a transition to the L_TPMPartial state.

LPM2: L_TPMSlumber	Transmit $PMREQ\_S_P$	
1. $PMACK_P$ received from Phy	→	L_ChkPhyRdy
2. X_RDY_P received from Phy	→	L_RcvWaitFifo[1]
3. $SYNC_P$ or R_OK_P received from Phy	→	L_TPMSlumber
4. AnyDword other than ($PMACK_P$ or X_RDY_P or $SYNC_P$ or R_OK_P or $PMREQ\_P_p$[3] or $PMREQ\_S_p$[3])[1] received from Phy layer	→	L_IDLE
5. $PMREQ\_P_p$ or $PMREQ\_S_p$ received from Phy	→	L_TPMSlumber[3]
6. PhyNRdy	→	L_NoCommErr[2]

NOTE –
1. This transition aborts the request from the Transport layer to enter a power mode. A status indication to the Transport layer of this event is required
2. This is an unexpected transition and constitutes an error condition. An error condition needs to be sent to the Transport layer as a result.
3. If $PMREQ\_P_p$ or $PMREQ\_S_p$ is received, the host shall make a transition to the L_IDLE state, but the device shall make a transition to the L_TPMSlumber state.

LPM3: L_PMOff	Transmit $PMACK_P$[1]	
1. 4 PMACK primitives sent	→	L_ChkPhyRdy
2. Less than four PMACK primitives sent	→	L_PMOff

NOTE –
1. A flag is set according to whether a $PMREQ\_P_P$ or $PMREQ\_S_P$ was received from the Phy layer

LPM4: L_PMDeny	Transmit $PMNACK_P$		
1.	$PMREQ\_P_P$ or $PMREQ\_S_P$ received from Phy	→	L_PMDeny
2.	AnyDword other than ($PMREQ\_P_P$ or $PMREQ\_S_P$) received from Phy layer	→	L_IDLE
3.	PhyNRdy	→	L_NoCommErr

LPM5: L_ChkPhyRdy	Assert Partial/Slumber to Phy (as appropriate)		
1.	PhyRdy	→	L_ChkPhyRdy
2.	PhyNRdy	→	L_NoCommPower

LPM6: L_NoCommPower	Maintain Partial/Slumber assertion (as appropriate)		
1.	Transport layer requests a wakeup or COMWAKE detected	→	L_WakeUp1
2.	Transport layer not requesting wakeup and COMWAKE not detected	→	L_NoCommPower

LPM7: L_WakeUp1	Deassert both Partial and Slumber		
1.	PhyRdy	→	L_WakeUp2
2.	PhyNRdy	→	L_WakeUp1

LPM8: L_WakeUp2	Transmit $ALIGN_P$		
1.	PhyRdy	→	L_IDLE
2.	PhyNRdy	→	L_NoCommErr

LPM1: L_TPMPartial state: This state is entered when the Transport layer has indicated that a transition to the Partial power state is desired.

Transition LPM1:1: When in this state a PMREQ_P primitive shall be transmitted. When the Link layer receives a PMACK primitive a transition to the LPM5: L_ChkPhyRdy state shall be made.

Transition LPM1:2: If the Link layer receives an X_RDY primitive a transition shall be made to the LR2: L_RcvWaitFifo state, effectively aborting the request to a power mode state.

Transition LPM1:3: If the Link layer receives a SYNC or R_OK primitive, then it is assumed that the opposite side has not yet processed the PMREQ_P primitive yet and time is needed. A transition to the LPM1: L_TPMPartial state shall be made.

Transition LPM1:4: If the host Link layer receives any Dword from the Phy layer other than a PMACK, X_RDY, SYNC or R_OK primitive, then the request to enter the partial state is aborted and a transition to L1: L_IDLE shall be made. This transition includes the case where a PMNACK is received. The device link layer shall not make this transition if it receives a PMREQ_P or PMREQ_S primitive while in this state.

Transition LPM1:5: The host Link layer shall not make this transition as it applies only to the device Link layer. If the device Link layer receives a PMREQ_P or PMREQ_S primitive from the host, it shall remain in this state by transitioning back to LPM1: L_TPMPartial.

Transition LPM1:6: If the Link layer detects that the Phy layer has become not ready, this is interpreted as an error condition. The Transport layer shall be notified of the condition and a transition shall be made to the LS1: L_NoCommErr state.

LPM2: L_TPMSlumber state: This state is entered when the Transport layer has indicated that a transition to the Slumber power state is desired.

Transition LPM2:1: When in this state a PMREQ_S primitive shall be transmitted. When the Link layer receives a PMACK primitive, a transition to the LPM5: L_ChkPhyRdy state shall be made.

Transition LPM2:2: If the Link layer receives an X_RDY primitive, a transition to the LR2: L_RcvWaitFifo state shall be made, effectively aborting the request to a power mode state.

Transition LPM2:3: If the Link layer receives a SYNC or R_OK primitive, then it is assumed that the opposite side has not yet processed the PMREQ_S primitive yet and time is needed. The transition to the LPM2: L_TPMSlumber state shall be made.

Transition LPM2:4: If the host Link layer receives any Dword from the Phy layer other than a PMACK, X_RDY, SYNC, or R_OK primitive, then the request to enter the slumber state is aborted and a transition to L1: L_IDLE shall be made. This transition includes the case where a PMNACK is received.. The device link layer shall not make this transition if it receives a PMREQ_P or PMREQ_S primitive while in this state.

Transition LPM2:5: The host Link layer shall not make this transition as it applies only to the device Link layer. If the device Link layer receives a PMREQ_P or PMREQ_S primitive from the host, it shall remain in this state by transitioning back to LPM2: L_TPMSlumber.

Transition LPM2:6: If the Link layer detects that the Phy layer has become not ready, this is interpreted as an error condition. The Transport layer shall be notified of the condition and a transition shall be made to the L_NoCommErr state.

LPM3: L_PMOff state: This state is entered when either a PMREQ_S or PMREQ_P primitive was received by the Link layer. The Link layer transmits a PMACK primitive for each execution of this state.

Transition LPM3:1: If four PMACK primitives have been transmitted, a transition shall be made to the L_ChkPhyRdy state.

Transition LPM3:2: If less than four PMACK primitives have been transmitted, a transition shall be made to L_PMOff state.

LPM4: L_PMDeny state: This state is entered when any primitive is received by the Link layer to enter a power mode and power modes are currently disabled. The Link layer shall transmit a PMNACK primitive to inform the opposite end that a power mode is not allowed.

Transition LPM4:1: If the Link layer continues to receive a request to enter any power mode than a transition back to the same LPM4: L_PMDeny state shall be made.

Transition LPM4:2: If the Link layer receives any Dword other than a power mode request primitive, then the Link layer assumes that the power mode request has been removed and shall make a transition to the L1: L_IDLE state.

Transition LPM4:3: If the Link layer detects that the Phy layer has become not ready, this is interpreted as an error condition. The Transport layer shall be notified of the condition and a transition shall be made to the LS1: L_NoCommErr state.

LPM5: L_ChkPhyRdy state: This state is entered whenever it is desired for the Phy layer to enter a low power condition. For each execution in this state a request is made to the Phy layer to enter the state and deactivate the PhyRdy signal. Partial or Slumber is asserted to the Phy as appropriate.

Transition LPM6:1: If the Phy layer has not yet processed the request to enter the power saving state and not deactivated the PhyRdy signal, then the Link layer shall remain in the LPM5: L_ChkPhyRdy state and continue to request the Phy layer to enter the power mode state.

Transition LPM6:2: When the Phy layer has processed the power mode request and has deactivated the PhyRdy signal, then a transition shall be made to the LPM6: L_NoCommPower state.

LPM6: L_NoCommPower state: This state is entered when the Phy has deasserted its PhyRdy signal indicating that it is in either Partial or Slumber state. In this state, the Link layer waits for the out of band detector to signal reception of the COMWAKE signal (for a wakeup initiated by the other device), or for the Transport layer to request a wakeup.

Transition LPM6:1: If the Transport layer requests a wakeup or the out of band signal detector indicates reception of the COMWAKE signal, then a transition shall be made to LPM7: L_WakeUp1

Transition LPM6:2: If the Transport layer does not request a wakeup and the out of band detector does not indicate reception of the COMWAKE signal, then a transition shall be made to LPM6: L_NoCommPower.

LPM7: L_WakeUp1 state: This state is entered when the Transport layer has initiated a wakeup. In this state, the Link layer shall deassert both Partial and Slumber to the Phy, and wait for the PhyRdy signal from the Phy to be asserted. While in this state the Phy is performing the wakeup sequence.

Transition LPM7:1 When the Phy asserts its PhyRdy signal, a transition shall be made to LPM8: L_WakeUp2.

Transition LPM7:2: When the Phy remains not ready, a transition shall be made to LPM7: L_WakeUp1.

LPM8: L_WakeUp2 state: This state is entered when the Phy has acknowledged an initiated wakeup request by asserting its PhyRdy signal. In this state, the Link layer shall transmit the ALIGN sequence, and transition to the L1: L_IDLE state.

Transition LPM8:1 If the Phy keeps PhyRdy asserted, a transition shall be made to the L1: L_IDLE state.

Transition LPM8:2 If the Phy deasserts PhyRdy, this is an error condition. The Transport layer shall be notified of the condition and a transition shall be made to the LS1: L_NoCommErr state.

8 Transport layer

8.1 Transport layer overview

The Transport layer need not be cognizant of how frames are transmitted and received. The Transport layer simply constructs Frame Information Structures (FIS's) for transmission and decomposes received Frame Information Structures. Host and device Transport layer state differ in that the source of the FIS content differs. The Transport layer maintains no context in terms of ATA commands or previous FIS content.

8.1.1 FIS construction

When requested to construct an FIS by a higher layer, the Transport layer provides the following services:

- Gathers FIS content based on the type of FIS requested.
- Places FIS content in the proper order.
- Notifies the Link layer of required frame transmission and passes FIS content to Link.
- Manages Buffer/FIFO flow, notifies Link of required flow control.
- Receives frame receipt acknowledge from Link layer.
- Reports good transmission or errors to requesting higher layer.

8.1.2 FIS decomposition

When an FIS is received from the Link layer, the Transport layer provides the following services:

- Receives the FIS from the Link layer.
- Determines FIS type.
- Distributes the FIS content to the locations indicated by the FIS type.
- For the host Transport layer, receipt of an FIS may also cause the construction of an FIS to be returned to the device.
- Reports good reception or errors to higher layer

8.2 Frame Information Structure (FIS)

8.3 Overview

A frame is a group of Dwords that convey information between host and device as described previously. Primitives are used to define the boundaries of the frame and may be inserted to control the rate of the information flow. This section will focus on and describe the actual information content of the frame - hereto referred as payload - and assumes the reader is aware of the Primitives that are needed to support the information content.

The contents of the info field is divided into three categories: (1) register type, (2) setup type, and (3) data type. For each category the organization of each frame is defined in the following section.

8.4 Payload content

The type and layout of the payload is indicated by the Frame Information Type field located in byte 0 of the first Dword of the payload. See Figure 59 – Register - Host to Device FIS layout as an example. This example type is used primarily to transfer the contents of the Shadow Register Block Registers from the host to the device. Table 26 can be referenced to refresh the reader's memory of a simplified version of the Shadow Register Block organization of a standard ATA adapter.

Table 26. – Simplified Shadow Register Block register numbering

				Register access operation	
	A2	**A1**	**A0**	**Read**	**Write**
CS 0 Active	0	0	0	Data Port	
	0	0	1	Error	Features
	0	1	0	Sector Count [15:8], [7:0]	
	0	1	1	Sector Number [31:24], [7:0]	
	1	0	0	Cylinder Low [39:32], [15:8]	
	1	0	1	Cylinder High [47:40], [23:16]	
	1	1	0	Device / Head	
	1	1	1	Status	Command
CS 1 active	1	1	0	Alternate Status	Device Control

The following sections will detail the types of payloads that are possible. The SOF, EOF and HOLD primitives have been removed for clarity.

8.5 FIS types

The following sections define the structure of each individual FIS.

8.5.1 All FIS types

In all of the following FIS structures the following rules shall apply:

1. All reserved fields shall be written or transmitted as all 0's
2. All reserved fields shall be ignored during the reading or reception process.

8.5.2 Register - Host to Device

0	Features	Command	C	R	R	Reserved (0)	FIS Type (27h)
1	Dev / Head	Cyl High	Cyl Low				Sector Number
2	Features (exp)	Cyl High (exp)	Cyl Low (exp)				Sector Num (exp)
3	Control	Reserved (0)	Sector Count (exp)				Sector Count
4	Reserved (0)	Reserved (0)	Reserved (0)				Reserved (0)

Figure 59 – Register - Host to Device FIS layout

Field Definitions

FIS Type - Set to a value of 27h. Defines the rest of the FIS fields. Defines the length of the FIS as five Dwords.

C - This bit is set to one when the register transfer is due to an update of the Command register. The bit is set to zero when the register transfer is due to an update of the Device Control register.

Command - Contains the contents of the Command register of the Shadow Register Block.

Sector Number - Contains the contents of the Sector Number register of the Shadow Register Block.

Control - Contains the contents of the Device Control register of the Shadow Register Block.

Cyl Low - Contains the contents of the Cylinder Low register of the Shadow Register Block.

Cyl Low (exp) – Contains the contents of the expanded address field of the Shadow Register Block

Cyl High - Contains the contents of the Cylinder High register of the Shadow Register Block.

Cyl High (exp) – Contains the contents of the expanded address field of the Shadow Register Block

Dev / Head - Contains the contents of the Device / Head register of the Shadow Register Block.

Features - Contains the contents of the Features register of the Shadow Register Block.

Features (exp) – Contains the contents of the expanded address field of the Shadow Register Block

R – Reserved bits – shall be set to 0.

Sector Count - Contains the contents of the Sector Count register of the Shadow Register Block.

Sector Count (exp) – Contains the contents of the expanded address field of the Shadow Register Block

Sector Number - Contains the contents of the Sector Number register of the Shadow Register Block.

Sector Num (exp) – Contains the contents of the expanded address field of the Shadow Register Block

8.5.2.1 Description

The Register – Host to Device FIS is used to transfer the contents of the Shadow Register Block from the host to the device. This is the mechanism for issuing legacy ATA commands to the device.

8.5.2.2 Transmission

Transmission of a Register – Host to Device FIS is initiated by a write operation to either the command register, or a write to the Control register with a value different than is currently in the Control register * in the host adapter's Shadow Register Block. Upon initiating transmission, the current contents of the Shadow Register Block are transmitted and the C bit in the FIS is set according to whether the transmission was a result of the Command register being written or the Control register being written. The host adapter shall set the BSY bit in the shadow Status register to one within 400 ns of the write operation to the Command register that initiated the transmission. The host adapter shall set the BSY bit in the shadow Status register to one within 400 ns of a write operation to the Control register if the write to the Control register changes the state of the SRST bit from 0 to 1 but shall not set the BSY bit in the shadow Status register for writes to the Control register that do not change the state of the SRST bit from 0 to 1.

It is important to note that Serial ATA host adapters enforce the same access control to the Shadow Register Block as legacy (parallel) ATA devices enforce to the Command Block Registers. Specifically, the host is prohibited from writing the Features, Sector Count, Sector Number, Cylinder Low, Cylinder High, or Device/Head registers when either BSY or DRQ is set in the Status Register. Any write to the Command Register when BSY or DRQ is set is ignored unless the write is to issue a Device Reset command.

8.5.2.3 Reception

Upon reception of a valid Register - Host to Device FIS the device updates its local copy of the Command and Control Block Register contents and either initiates execution of the command indicated in the command register or initiates execution of the control request indicated in the Control register, depending on the state of the C bit in the FIS.

* There are legacy BIOS and drivers that will write the Control register to enable the interrupt just prior to issuing a command. To avoid unnecessary overhead, this FIS is transmitted to the device only upon a change of state from the previous value.

8.5.3 Register - Device to Host

0	Error	Status	R	I	R	Reserved (0)	FIS Type (34h)
1	Dev / Head	Cyl High	Cyl Low				Sector Number
2	Reserved (0)	Cyl High (exp)	Cyl Low (exp)				Sector Num (exp) (0)
3	Reserved (0)	Reserved (0)	Sector Count (exp)				Sector Count
4	Reserved (0)	Reserved (0)	Reserved (0)				Reserved (0)

Figure 60 – Register - Device to Host FIS layout

Field Definitions

FIS Type - Set to a value of 34h. Defines the rest of the FIS fields. Defines the length of the FIS as five Dwords.

Cyl Low - Contains the new value of the cylinder low register of the Shadow Register Block.

Cyl Low (exp) – Contains the contents of the expanded address field of the Shadow Register Block

Cyl High - Contains the new value of the cylinder high register of the Shadow Register Block.

Cyl High (exp) – Contains the contents of the expanded address field of the Shadow Register Block

Dev / Head - Contains the new value of the device / head register of the Shadow Register Block.

Error - Contains the new value of the Error register of the Shadow Register Block.

I - Interrupt bit. This bit reflects the interrupt bit line of the device

R - Reserved (0)

Sector Count - Contains the new value of the sector count register of the Shadow Register Block.

Sector Count (exp) – Contains the contents of the expanded address field of the Shadow Register Block

Sector Number - Contains the new value of the sector number register of the Shadow Register Block.

Sector Num (exp) – Contains the contents of the expanded address field of the Shadow Register Block

Status - Contains the new value of the Status (and Alternate status) register of the Shadow Register Block.

8.5.3.1 Description

The Register – Device to Host FIS is used to by the device to update the contents of the host adapter's Shadow Register Block. This is the mechanism by which devices indicate command completion status or otherwise change the contents of the host adapter's Shadow Register Block.

8.5.3.2 Transmission

Transmission of a Register - Device to Host FIS is initiated by the device in order to update the contents of the host adapter's Shadow Register Block. Transmission of the Register - Device to Host FIS is typically as a result of command completion by the device.

The Register - Device to Host FIS should not be used to set the SERV bit in the Status Register to request service for a bus released command; the Set Device Bits - Device to Host FIS should be used instead. The SERV bit transmitted with the Register - Device to Host FIS will be written to the shadow Status Register and so the bit should accurately reflect the state of pending service requests when the FIS is transmitted as a result of a command completion by the device.

8.5.3.3 Reception

Upon reception of a valid Register - Device to Host FIS the received register contents are transferred to the host adapter's Shadow Register Block.

If the BSY bit and DRQ bit in the shadow Status Register are both cleared when a Register - Device to Host FIS is received by the host adapter, then the host adapter shall discard the contents of the received FIS and not update the contents of any shadow register.

8.5.4 Set Device Bits - Device to Host

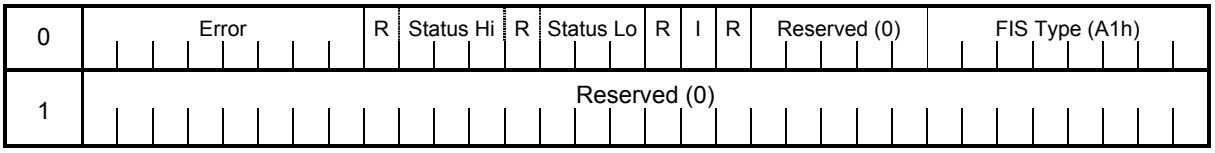


Figure 61 - Set Device Bit - Device to Host FIS layout

Field Definitions

FIS Type – Set to a value of A1h. Defines the rest of the FIS fields. Defines the length of the FIS as two Dwords.

I – Interrupt Bit. This bit signals the host adapter to enter an interrupt pending state if both the BSY bit and the DRQ bit in the shadow Status register are zero when the frame is received.

Error – Contains the new value of the Error register of the Shadow Register Block.

Status-Hi – Contains the new value of bits 6, 5, and 4 of the Status register of the Shadow Register Block.

Status-Lo – Contains the new value of bits 2,1, and 0 of the Status register of the Shadow Register Block.

R – Reserved (0)

8.5.4.1 Description

The Set Device Bits - Device to Host FIS is used by the device to load Shadow Register Block bits for which the device has exclusive write access. These bits are the eight bits of the Error register and six of the eight bits of the Status register. This FIS does not alter bit 7, BSY, or bit 3, DRQ, of the Status register.

The FIS includes a bit to signal the host adapter to generate an interrupt if the BSY bit and the DRQ bit in the shadow Status Register are both cleared to zero when this FIS is received.

Some Serial ATA to parallel ATA bridge solutions may elect to not support this FIS based on the requirements of their target markets. Upon reception, such devices will process this FIS as if it were an invalid FIS type and return the R_ERR end of frame handshake.

8.5.4.2 Transmission

The device transmits a Set Device Bits - Device to Host to alter one or more bits in the Error register or in the Status register in the Shadow Register Block. This FIS should be used by the device to set the SERV bit in the Status register to request service for a bus released command. When used for this purpose the device shall set the Interrupt bit to one.

8.5.4.3 Reception

Upon receiving a Set Device Bits - Device to Host, the host adapter shall load the data from the Error field into the shadow Error register, the data from the Status-Hi field into bits 6, 5, and 4, of the shadow Status register, and the data from the Status-Lo field into bits 2, 1, and 0 of the shadow Status register. Bit 7, BSY, and bit 3, DRQ, of the shadow Status register shall not be changed. If the I bit in the FIS is set to a one, and if both the BSY bit and the DRQ bit in the Shadow status register are cleared to zero when this FIS is received, then the host adapter shall enter an interrupt pending state.

8.5.5 DMA Activate - Device to Host

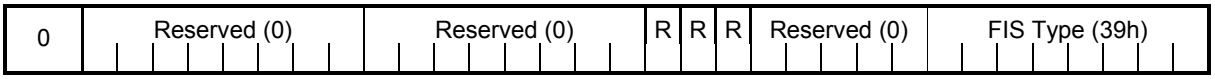


Figure 62 – DMA Activate - Device to Host FIS layout

Field Definitions

FIS Type - Set to a value of 39h. Defines the rest of the FIS fields. Defines the length of the FIS as one Dword.

R - Reserved (0)

8.5.5.1 Description

The DMA Activate – Device to Host FIS is used by the device to signal the host to proceed with a DMA data transfer of data from the host to the device. This is the mechanism by which a legacy device signals its readiness to receive DMA data from the host.

A situation may arise where the host needs to send multiple Data FIS's in order to complete the overall data transfer request. The host shall wait for a successful reception of a DMA Activate FIS before sending each of the Data FIS's that are needed.

8.5.5.2 Transmission

The device transmits a DMA Activate – Device to Host to the host in order to initiate the flow of DMA data from the host to the device as part of the data transfer portion of a corresponding DMA write command. When transmitting this FIS, the device shall be prepared to subsequently receive a Data - Host to Device FIS from the host with the DMA data for the corresponding command.

8.5.5.3 Reception

Upon receiving a DMA Activate – Device to Host, if the host adapter's DMA controller has been programmed and armed, the host adapter shall initiate the transmission of a Data FIS and shall transmit in this FIS the data corresponding to the host memory regions indicated by the DMA controller's context. If the host adapter's DMA controller has not yet been programmed and armed, the host adapter shall set an internal state indicating that the DMA controller has been activated by the device, and as soon as the DMA controller has been programmed and armed, a Data FIS shall be transmitted to the device with the data corresponding to the host memory regions indicated by the DMA controller context.

8.5.6 DMA Setup – Device to Host or Host to Device (Bidirectional)

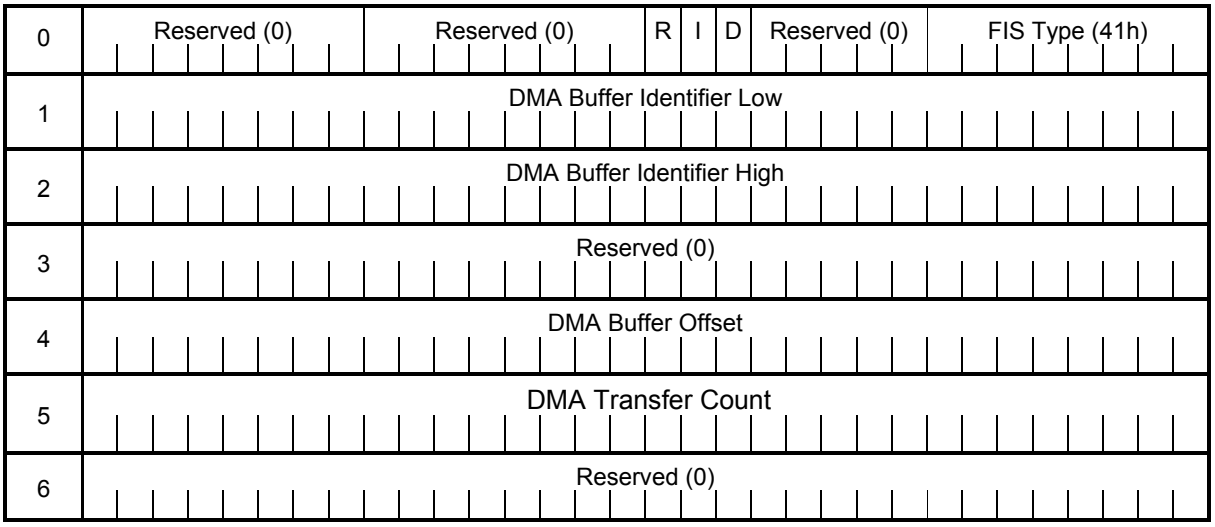


Figure 63 –DMA Setup – Device to Host FIS layout

Field Definitions

FIS Type - Set to a value of 41h. Defines the rest of the FIS fields. Defines the total length of the FIS as seven Dwords.

D - Indicates whether subsequent data transferred after this FIS is from transmiter to receiver or from receiver to transmitter 1 = transmitter to receiver, 0 = receiver to transmitter.

DMA Buffer Identifier Low/High - This field is used to identify a DMA buffer region in host memory. The contents are not described in this specification and are host dependent. The buffer identifier is supplied by the host to the device and the device echoes it back to the host. This allows the implementation to pass a physical address or or, in more complex implementations, the buffer identifier could be a scatter gather list or other information that can identify a DMA “channel”.

DMA Buffer Offset - This is the byte offset into the buffer. Bits <1:0> must be zero

DMA Transfer Count: This is the number of bytes that will be read or written by the device. Bit zero must be zero.

I Interrupt - If the Interrupt bit is set to 1 an interrupt pending shall be generated when the DMA transfer count is exhausted.

R - Reserved (0)

8.5.6.1 Description

The DMA Setup – Device to Host or Host to Device FIS is the mechanism by which first-party DMA access to host memory is initiated. This FIS is used to request the host or device to program its DMA controller before transferring data. The FIS allows the actual host memory regions to be abstracted (depending on implementation) by having memory regions referenced via a base memory descriptor representing a memory region that the host has granted the device access to. The specific implementation for the memory descriptor abstraction is not defined.

The device or host is informed of the 64-bit DMA buffer identifier/descriptor at some previous time by an implementation specific mechanism such as a command issued to or as defined in a specification. Random access within a buffer is accomplished by using the buffer offset.

First party DMA is a superset capability not necessarily supported by legacy devices or legacy device drivers but essential for accommodating future capabilities.

8.5.6.2 Transmission

A device or host transmits a DMA Setup – Device to Host or Host to Device FIS as the first step in performing a DMA access. The purpose of the DMA Setup – Device to Host or Host to Device is to establish DMA hardware context for one or more data transfers.

A DMA Setup – Device to Host or Host to Device is required only when the DMA context is to be changed. Multiple Data – Host to Device or Device to Host FIS's can follow in either direction, for example, if the transfer count exceeds the maximum Data – Host to Device or Device to Host transfer length or when a data transfer is interrupted. When multiple Data – Host to Device or Device to Host FIS's follow a DMA Setup – Device to Host or Host to Device FIS, the device or host shall place the data contained in the FIS in sequential addresses; that is, if the last Dword of an FIS is placed in (or obtained from) address N, the first Dword of a subsequent Data – Host to Device or Device to Host shall be placed in (or obtained from) address N+4 unless an intervening DMA Setup – Device to Host or Host to Device FIS is used to alter the DMA context. This mechanism allows for the efficient streaming of data into a buffer.

8.5.6.3 Reception

Upon receiving a DMA Setup – Device to Host or Host to Device FIS, the receiver of the FIS shall validate the received DMA Setup request, and provided that the buffer identifier and the specified offset/count are valid, program and arm the adapter's DMA controller using the information in the FIS. The specific implementation of the buffer identifier and buffer/address validation is not specified. After a valid DMA Setup – Device to Host or Host to Device FIS with the D bit set to 0, the receiver of the DMA Setup – Device to Host or Host to Device FIS responds with one or more Data – Host to Device or Device to Host FIS's until the DMA count is exhausted. After a valid DMA Setup – Device to Host or Host to Device FIS with the D bit set to 1, the receiver of the FIS must be prepared to accept one or more Data – Host to Device or Device to Host FIS's until the DMA count is exhausted.

An interrupt pending condition shall be generated upon the completion of the DMA transfer if the I bit is set to 1. The definition DMA transfer completion is system dependent but typically includes the exhaustion of the transfer count or the detection of an error by the DMA controller.

NOTE – First-party DMA accesses are categorized in two groups: command/status transfers and user-data transfers. Interrupts would not typically be generated on user-data transfers. The optimal interrupt scheme for command/status transfers is not defined in this specification.

8.5.7 BIST Activate - Bidirectional

0	Reserved (0)								Pattern Definition								R	R	R	Reserved (0)								FIS Type (58h)							
									T	A	S	L	F	P	R	V																
1	Data [31:24]								Data [23:16]								Data [15:8]								Data [7:0]							
2	Data [31:24]								Data [23:16]								Data [15:8]								Data [7:0]							

Figure 64 – BIST Activate - Bidirectional

Field Definitions

FIS Type - Set to a value of 58h. Defines the rest of the FIS fields.
F – Far End Analog (AFE) Loopback (Optional)
L - Far End Retimed Loopback* Transmitter must insert additional ALIGNS
R - Reserved (0)
T - Far end transmit only mode
A - ALIGN Bypass (Do not Transmit Align Primitives) (valid only in combination with T Bit) (optional behavior)
S - Bypass Scrambling (valid only in combination with T Bit) (optional behavior)
P - Primitive bit. (valid only in combination with the T- Bit) (optional behavior)

V - Vendor Unique Test Mode. Causes all other bits to be ignored

8.5.7.1 Description

The BIST Activate FIS shall be used to place the receiver in one of “n” loopback modes.

The BIST Activate FIS is a bi-directional request in that it can be sent by either the host or the device. The sender and receiver have distinct responsibilities in order to insure proper cooperation between the two parties. The state machines for transmission and reception of the FIS are symmetrical.

The state machines for the transmission of the FIS do not attempt to specify the actions the sender takes once successful transmission of the request has been performed. After the Application layer is notified of the successful transmission of the FIS the sender’s Application layer will prepare its own Application, Transport and Physical layers into the appropriate states that support the transmission of a stream of data. The FIS shall not be considered successfully transmitted until the receiver has acknowledged reception of the FIS as per normal FIS transfers documented in various sections of this specification. The transmitter of the BIST Activate FIS should transmit continuous SYNC primitives after reception of the R_OK primitive until such a time that it is ready to interact with the receiver in the BIST exchange.

Similarly, the state machines for the reception of the FIS do not specify the actions of the receiver’s application layer. Once the FIS has been received, the receiver’s application layer must place its own Application, Transport and Physical layers into states that will perform the appropriate retransmission of the sender’s data. The receiver shall not enter the BIST state until after it has properly received a good BIST Activate FIS (good CRC), indicated a successful transfer of the FIS to the transmitting side via the R_OK primitive and has received at least one good SYNC primitive. Once in the self-test mode, a receiver shall continue to allow processing of the COMINIT or COMRESET signals in order to exit from the self-test mode.

F: The Far End Analog (Analog Front End - AFE) Loopback, shall be defined as a vendor optional mode where the raw data is received, and retransmitted, without any retiming or re-synchronization, etc. The implementation of Far End AFE Loopback is optional due to the round-trip characteristics of the test as well as the lack of retiming. This mode is intended to give a quick indication of connectivity, and test failure is not an indication of system failure.

L: The Far End Retimed Loopback, shall be defined as a mode where the receiver will retime the data, and retransmit the retimed data. The initiator of the retimed loopback mode must account for the loopback device consuming up to two ALIGN primitives (one ALIGN sequence) every 256 Dwords transmitted and, if it requires any ALIGN primitives to be present in the returned data stream, it must insert additional ALIGN's in the transmitted stream. The initiator shall transmit additional ALIGN sequences in a single burst at the normal interval of every 256 Dwords transmitted (as opposed to inserting ALIGN sequences at half the interval).

The loopback device may remove zero, one, or two ALIGN primitives from the received data. It may insert one or more ALIGN primitives if they are directly preceded or followed by the initiator inserted ALIGN primitives (resulting in ALIGN sequences consisting of at least two ALIGN primitives) or it may insert two or more ALIGN primitives if not preceded or followed by the initiator's ALIGN primitives. One side effect of the loopback retiming is that the returned data stream may have instances of an odd number of ALIGN primitives, however, returned ALIGN's are always in bursts and if the initiator transmitted dual ALIGN sequences (four consecutive ALIGN's), then the returned data stream shall include ALIGN bursts that are no shorter than two ALIGN primitives long (although the length of the ALIGN burst may be odd). The initiator of the retimed loopback mode shall not assume any relationship between the relative position of the ALIGN's returned by the loopback device and the relative position of the ALIGN's sent by the initiator.

In retimed loopback mode, the initiator shall transmit only valid 8b/10b characters so the loopback device may 10b/8b decode it and re-encode it before retransmission. If the loopback device descrambles incoming data it is responsible for rescrambling it with the same sequence of scrambling syndromes in order to ensure the returned data is unchanged from the received data. The loopback device's running disparity for its transmitter and receiver are not guaranteed to be the same and thus the loopback initiator shall 10b/8b decode the returned data rather than use the raw 10b returned stream for the purpose of data comparison. The loopback device shall return all received data unaltered and shall disregard protocol processing of primitives. Only the OOB signals and ALIGN processing is acted on by the loopback device, while all other data is retransmitted without interpretation.

T: The Far-End Transmit Mode shall be defined as a mode that can be used to invoke the Far-End Interface to send data patterns, upon receipt of the FIS BIST Activate, as defined by Pattern DWords #1, and #2. Note that Pattern D-Words #1, and #2 shall be applicable only when the T bit is active, indicating "Far-End Transmit Mode".
This data is modified by the following bits.

Note, that this mode is intended for Inspection/Observation Testing, as well as support for conventional laboratory equipment, rather than for in-system automated testing.

P: The transmit primitives bit. When this bit is set in far end transmit mode, the lowest order byte of the two following Dwords will be treated as K Characters. It is the responsibility of the sender of the BIST FIS to ensure that the data in byte 0 of the Dwords are valid D character versions of the K character (i.e. BCh for K28.5). Applicable only when the T bit is set.

A: The ALIGN primitive sequence bypass mode. When set to 1, no ALIGN primitives are sent. When the A-bit is not asserted, ALIGN Primitives are sent normally as defined in this document. Applicable only when the T bit is set.

S: The Bypass Scrambling shall be defined as a mode that can be used to send data or patterns, during BIST activation, that are not scrambled, however are encoded and decoded to normal and legal 8b/10b values. Applicable only when the T bit is set.

V: The vendor unique mode is implementation specific and shall be reserved for individual vendor use. All other bits are ignored in this mode.

8.5.7.2 Transmission

The initiator transmits a BIST Activate to the recipient in order to initiate the BIST mode of operation.

8.5.7.3 Reception

Upon receiving a BIST Activate, the recipient shall begin operations as per the BIST Activate FIS, described in this document above.

8.5.8 PIO Setup – Device to Host

0	Error							Status					R	I	D	Reserved (0)											FIS Type (5Fh)							
1	Dev / Head							Cyl High								Cyl Low											Sector Number							
2	Reserved (0)							Cyl High (exp)								Cyl Low (exp)											Sector Num (exp) (0)							
3	E_Status							Reserved (0)								Sector Count (exp)											Sector Count							
4	Reserved (0)															Transfer Count																		

Figure 65 – PIO Setup - Device to Host FIS layout

Field Definitions

FIS Type - Set to a value of 5Fh. Defines the rest of the FIS fields. Defines the length of the FIS as five Dwords.

Cyl Low - Holds the contents of the cylinder low register of the Commanf Block.

Cyl Low (exp) – Contains the contents of the expanded address field of the Shadow Register Block

Cyl High - Holds the contents of the cylinder high register of the Command Block.

Cyl High (exp) – Contains the contents of the expanded address field of the Shadow Register Block

D - Indicates whether host memory is being written or read by the device. 1 = write (device to host), 0 = read (host to device).

Dev / Head - Holds the contents of the device / head register of the Command Block.

Status - Contains the new value of the Status register of the Command Block for initiation of host data transfer.

Error - Contains the new value of the Error register of the Command Block at the conclusion of all subsequent Data to Device frames.

I - Interrupt bit. This bit reflects the interrupt bit line of the device

R - Reserved (0)

Sector Count - Holds the contents of the sector count register of the Command Block.

Sector Count (exp) – Contains the contents of the expanded address field of the Shadow Register Block

Sector Number - Holds the contents of the sector number register of the Command Block.

Sector Num (exp) – Contains the contents of the expanded address field of the Shadow Register Block

E_Status - Contains the new value of the Status register of the Command Block at the conclusion of all subsequent Data frames.

Transfer Count – Holds the number of bytes to be transferred in the subsequent data FIS. The Transfer Count value shall be nonzero and the low order bit shall be zero (even number of bytes transferred).

8.5.8.1 Description

The PIO Setup – Device to Host FIS is used by the device to provide the host adapter with sufficient information regarding a PIO data phase to allow the host adapter to efficiently handle PIO data transfers. For PIO data transfers, the device shall send to the host a PIO Setup – Device to Host FIS just before each and every data transfer FIS that is required to complete the data transfer. Data transfers from Host to Device as well as data transfers from Device to Host shall follow this algorithm. Because of the stringent timing constraints in the ATA standard, the PIO Setup FIS includes both the starting and ending status values. These are used by the host adapter to first signal to host software readiness for PIO write data (BSY deasserted and DRQ asserted), and following the PIO write burst to properly signal host software by deasserting DRQ and possibly raising BSY.

8.5.8.2 Transmission of PIO Setup by Device Prior to a Data Transfer from Host to Device

The device transmits a PIO Setup – Device to Host FIS to the host in preparation for a PIO data payload transfer just before each and every PIO data payload transfer required to complete the total data transfer for a command. The device includes in the FIS the values to be placed in the Status shadow register at the beginning of the PIO data payload transfer and the value to be placed in the Status shadow register at the end of the data payload transfer. The device must be prepared to receive a Data FIS in response to transmitting a PIO Setup FIS.

8.5.8.3 Reception of PIO Setup by Host Prior to a Data Transfer from Host to Device

Upon receiving a PIO Setup – Device to Host FIS, the host shall transfer the Status and Error values into the Status and Error Shadow registers and shall hold the E_Status value in a temporary register. The Transfer Length value shall be loaded into a countdown register. Upon detecting the change in the Status shadow register, host software proceeds to perform a series of write operations to the Data shadow register, which the host adapter shall collect to produce a Data FIS to the device. Each write of the Data shadow register results in another word of data being concatenated into the Data FIS, and the countdown register being decremented accordingly. The E_Status value shall be transferred to the Status shadow register within 400 ns of the countdown register reaching terminal count. In the case that the transfer length represents an odd number of words, the last word shall be placed in the low order (word 0) of the final Dword and the high order word (word 1) of the final Dword shall be padded with zeros before transmission. This process is repeated for each and every data FIS needed to complete the overall data transfer of a command.

8.5.8.4 Transmission of PIO Setup by Device Prior to a Data Transfer from Device to Host

The device transmits a PIO Setup – Device to Host FIS to the host in preparation for a PIO data payload transfer just before each and every PIO data payload transfer required to complete the total data transfer for a command. The device includes in the FIS the values to be placed in the Status shadow register at the beginning of the PIO data payload transfer and the value to be placed in the Status shadow register at the end of the data payload transfer. The device must be prepared to transmit a Data FIS in following the transmittal of a PIO Setup FIS.

8.5.8.5 Reception of PIO Setup by Host Prior to a Data Transfer from Device to Host

Upon receiving a PIO Setup – Device to Host FIS for a device to host transfer, the host shall hold the Status, Error, and E_Status values in temporary registers. The Transfer Length value shall be loaded into a countdown register. Upon reception of a Data FIS from the device, the Status and Error values are loaded into the Status and Error Shadow registers and host software proceeds to perform a series of read operations from the Data shadow register. Each read of the Data shadow register results in a countdown register being decremented accordingly. The E_Status value shall be transferred to the Status shadow register within 400 ns of the countdown register reaching terminal count. This process is repeated for each and every data FIS needed to complete the overall data transfer of a command.

8.5.9 Data - Host to Device or Device to Host (Bidirectional)

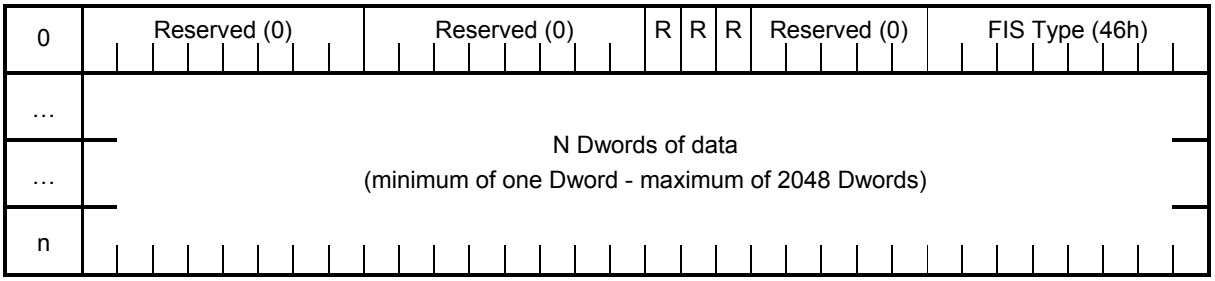


Figure 66 – Data – Host to Device or Device to Host FIS layout

Field Definitions

FIS Type - Set to a value of 46h. Defines the rest of the FIS fields. Defines the length of the FIS as n+1 Dwords.

Dwords of data - Contain the actual data to transfer. Only 32 bit fields are transferred. The last Dword is padded with zeros when only a partial Dword is to be transmitted.

NOTE –The maximum amount of user data that can be sent in a single Data – Host to Device or Data – Device to Host FIS is limited. See description.

R – Reserved bit – shall be set to 0.

8.5.9.1 Description

The Data – Host to Device and the Data – Device to Host FIS's are used for transporting payload data, such as the data read from or written to a number of sectors on a hard drive. The FIS may either be generated by the device to transmit data to the host or may be generated by the host to transmit data to the device. This FIS is generally only one element of a sequence of transactions leading up to a data transmission and the transactions leading up to and following the Data FIS establish the proper context for both the host and device.

The byte count of the payload is not an explicit parameter, rather it is inferred by counting the number of Dwords between the SOF and EOF primitives, and discounting the FIS type and CRC Dwords. The payload size shall be no more than 2048 Dwords (8192 bytes) for native implementations. Parallel ATA to Serial ATA bridge implementations should not transmit more than 2048 Dwords (8192 bytes) in any one Data FIS. However, separate bridge hardware is not required to break up PIO or DMA transfers from parallel ATA into more than one Serial ATA Data FIS and may therefore produce payloads longer than 2048 Dwords. Receivers shall tolerate reception of data payloads longer than

2048 Dwords without error. Non-packet devices, with or without bridges, should report a SET MULTIPLE limit of 16 sectors or less in word 47 of their IDENTIFY DEVICE information.

In the case that the transfer length represents an odd number of words, the last word shall be placed in the low order (word 0) of the final Dword and the high order word (word 1) of the final Dword shall be padded with zeros before transmission.

8.5.9.2 Transmission

The device transmits a Data – Device to Host FIS to the host during the data transfer phase of legacy mode PIO reads, DMA reads, and First-party DMA writes to host memory. The device shall precede a Data FIS with any necessary context-setting transactions as appropriate for the particular command sequence. For example, a First-party DMA host memory write must be preceded by a First-party DMA Setup – Device to Host FIS to establish proper context for the Data FIS that follows.

The host transmits a Data – Host to Device FIS to the device during the data transfer phase of PIO writes, DMA writes, and First-party DMA reads of host memory. The FIS shall be preceded with any necessary context-setting transactions as appropriate for the particular command sequence. For example, a legacy mode DMA write to the device is preceded by a DMA Activate – Device to Host FIS with the DMA context having been pre-established by the host.

When used for transferring data for DMA operations multiple Data – Host to Device or Device to Host FIS's can follow in either direction. Segmentation can occur when the transfer count exceeds the maximum Data – Host to Device or Device to Host transfer length or if a data transfer is interrupted.

When used for transferring data in response to a PIO Setup all of the data must be transmitted in a single Data FIS.

In the event that a transfer is broken into multiple FIS's, all intermediate FIS's must contain an integral number of full Dwords. If the total data transfer is for an odd number of words, then the high order word (word 1) of the last Dword of the last FIS shall be padded with zeros before transmission and discarded on reception.

The Serial ATA protocol does not permit for the transfer of an odd number of bytes.

8.5.9.3 Reception

Neither the host nor device is expected to buffer an entire Data FIS in order to check the CRC of the FIS before processing the data. Incorrect data reception for a Data FIS should be reflected in the overall command completion status.

8.6 Host transport states

8.6.1 Host transport idle state diagram

HTI1: HT_HostIdle	Host adapter waits for frame or frame request.	
1. Shadow Command register written	→	HT_CmdFIS
2. Shadow Device Control register written	→	HT_CntrlFIS [1]
3. Frame receipt indicated by Link layer	→	HT_ChkTyp
4. First party DMA host to device communication indicated by application layer	→	HT_DMASTUPFIS
5. BIST FIS Transmission requested	→	HT_XmitBIST
6. Previous FIS was PIO Setup and Application layer indicates data direction is host to device	→	HT_PIOOTrans2 [2]
NOTE: 1. The transition to this state is mandatory if the state of the SRST bit in the Device Control Shadow Register is changed, and is optional if the state of the SRST bit is not changed. 2. The PIO Setup FIS shall set an indication that PIO Setup was the last FIS received. Indication from the application layer that it is transmitting data to the device can be determined from the application layer performing write operations to the Data register in the Shadow Register Block.		

HTI2: HT_ChkTyp	Received FIS type checked.	
1. Register FIS type detected	→	HT_RegFIS
2. Set Device Bits FIS type detected	→	HT_DB_FIS
3. DMA Activate FIS type detected	→	HT_DMA_FIS
4. PIO Setup FIS type detected)	→	HT_PS_FIS
5. First Party DMA Setup FIS type detected	→	HT_DS_FIS
6. BIST FIS type detected	→	HT RcvBIST
7. Data FIS type detected and previous FIS was not PIO Setup	→	HT_DMAITrans [1]
8. Data FIS type detected and previous FIS was PIO Setup	→	HT_PIOITrans1 [1]
9. Unrecognized FIS received	→	HT_HostIdle
10. Notification of illegal transition error received from Link layer	→	HT_HostIdle
NOTE: 1. The PIO Setup FIS shall set an indication that PIO Setup was the last FIS sent, so that this state can determine whether to transition to DMA data transfer, or PIO data transfer.		

HTI1: HT_HostIdle state: This state is entered when a Frame Information Structure (FIS) transaction has been completed by the Transport layer.

When in this state, the Transport layer waits for the shadow Command register to be written, the shadow Device Control register to be written, or the Link layer to indicate that an FIS is being received.

TransitionHTI1:1: When the shadow Command register is written, the Transport layer shall make a transition to the HTCM1: HT_CmdFIS state.

Transition HTI1:2: When the shadow Control register is written, and the state of the SRST bit is changed from its previous state, the Transport layer shall make a transition to the HTCR1: HT_CntrlFIS state. This transition may optionally be made upon a write operation to the Control shadow register, or upon a write operation to the Control shadow register that changes its value from the previous value.

Transition HTI1:3: When the Link layer indicates that an FIS is being received, the Transport layer shall make a transition to the HTI2: HT_ChkTyp state.

Transition HTI1:4: When the Application layer indicates that a DMA Setup FIS is to be sent, the Transport layer shall make a transition to the HT_DMASTUP0:HT_DMASTUPFIS state.

Transition HTI1:5: When the application layer requests the transmission of a BIST request to the device the Transport layer shall make a transition to the HTXBIST1:HT state.

Transition HTI1:6: When the application layer requests the transmission of data to the device and the previous FIS was a PIOSetup type, the Transport layer shall make a transition to the HTPS3:HT_PIOOTrans2 state. The application layer signals transmission of PIO data to the device by performing writes to the Data register in the Shadow Register Block.

HTI2: HT_ChkTyp state: This state is entered when the Link layer indicates that an FIS is being received.

When in this state, the Transport layer checks the FIS type of the incoming FIS.

Transition HTI2:1: When the incoming FIS is a register type, the Transport layer shall notify the Link Layer that it has received a valid FIS, and make a transition to the HTR1: HT_RegFIS state.

Transition HTI2:2: When the incoming FIS is a Set Device Bits type, the transport layer shall notify the Link Layer that it has received a valid FIS and make a transition to the HTDB0:HT_DB_FIS state.

Transition HTI2:3: When the incoming FIS is a DMA Activate type, the Transport layer shall notify the Link Layer that it has received a valid FIS, and make a transition to the HTDA1: HT_DMA_FIS state.

Transition HTI2:4: When the incoming FIS is a PIO Setup type, the Transport layer shall notify the Link Layer that it has received a valid FIS, and make a transition to the HTPS1: HT_PS_FIS state.

Transition HTI2:5: When the incoming FIS is a First-party DMA Setup type, the Transport layer shall notify the Link Layer that it has received a valid FIS, and make a transition to the HTDS1: HT_DS_FIS state.

Transition HTI2:6: When the incoming FIS is a BIST Activate type, the Transport layer shall notify the Link Layer that it has received a valid FIS, and make a transition to the HTRBIST1:HT_RcvBIST state.

Transition HTI2:7: When the incoming FIS is a Data type, and the previous FIS was not a PIO Setup type, the Transport layer shall notify the Link Layer that it has received a valid FIS, and make a transition to the HTDA6:HT_DMAITrans state.

Transition HTI2:8: When the incoming FIS is a Data type, and the previous FIS was a PIO Setup type, the Transport layer shall notify the Link Layer that it has received a valid FIS, and make a transition to the HTPS6:HT_PIOITrans state.

Transition HTI2:9: When the received FIS is of an unrecognized, or unsupported type, the Transport layer shall notify the Link Layer that it has received an unrecognized FIS, and make a transition to the HTI1: HT_HostIdle state.

Transition HTI2:10: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1: HT_HostIdle state.

8.6.2 Host Transport transmit command FIS diagram

This protocol builds an FIS that contains the host adapter shadow register content and sends it to the device when the software driver or BIOS writes the host adapter shadow Command register.

HTCM1: HT_CmdFIS	Construct Cmd type FIS from the content of the shadow registers and notify Link to transfer.	
1. FIS transfer complete	→	HT_TransStatus
2. Notification of illegal transition error received from Link layer	→	HT_HostIdle
3. Frame receipt indicated by Link layer	→	HT_HostIdle

HTCM2: HT_TransStatus	Check Link and Phy transmission results and if an error occurred take appropriate action.	
1. Status checked, and no error detected	→	HT_HostIdle
2. Status checked, and error detected	→	HT_CmdFIS
3. Status checked, error detected, and SRST asserted	→	HT_HostIdle
4. Notification of illegal transition error received from Link layer	→	HT_HostIdle

HTCM1: HT_CmdFIS state: This state is entered when the shadow Command register is written.

When in this state, the Transport layer shall construct a register FIS, notify the Link layer that the FIS is to be transmitted, and pass the FIS to the Link layer. The Transport layer shall not perform flow control on a register FIS.

Transition HTCM1:1: When the entire FIS has been passed to the Link layer, the Transport layer shall indicate to the Link layer that the FIS transmit is complete and make a transition to the HTCM2: HT_TransStatus state.

Transition HTCM1:2: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1: HT_HostIdle state.

Transition HTCM1:3: When the Link layer indicates that an FIS is being received, the Transport Layer shall make a transition to the HTI1:HT_HostIdle state.

HTCM2: HT_TransStatus state: This state is when the entire FIS has been passed to the Link layer.

When in this state, the Transport layer shall wait for the Link and Phy ending status for the FIS and take appropriate error handling action if required.

Transition HTCM2:1: When the FIS status has been handled and no errors detected, the Transport layer shall transition to the HTI1: HT_HostIdle state.

Transition HTCM2:2: When the FIS status has been handled and an error has been detected, and the host has not asserted the SRST by writing to the Device Control register, the Transport layer shall transition to the HTCM1: HT_CmdFIS state.

Transition HTCM2:3: When the FIS status has been handled, an error has been detected, and the host has asserted the SRST by writing to the Device Control register, or a device reset command has

been written to an ATAPI device, the Transport layer shall inform the Link layer to send a SYNC primitive, and transition to the HTCM1: HT_HostIdle state.

Transition HTCM2:4: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1: HT_HostIdle state.

8.6.3 Host Transport transmit control FIS diagram

This protocol builds an FIS that contains the host adapter shadow register content and sends it to the device when the software driver or BIOS writes the host adapter shadow Device Control register.

HTCR1: HT_CntrlFIS	Construct Cntrl type FIS from the content of the shadow registers and notify Link to transfer.	
1. FIS transfer complete	→	HT_TransStatus
2. Notification of illegal transition error received from Link layer	→	HT_HostIdle
3. Frame receipt indicated by Link layer	→	HT_HostIdle

HTCR2: HT_TransStatus	Check Link and Phy transmission results and if an error occurred take appropriate action.	
1. Status checked and no errors detected	→	HT_HostIdle
2. Status check and at least one error detected	→	HT_Cntrl_FIS

HTCR1: HT_Cntrl_FIS state: This state is entered when the shadow Device Control register is written.

When in this state, the Transport layer shall construct a register FIS, notify the Link layer that the FIS is to be transmitted, and pass the FIS to the Link layer. The Transport layer shall not perform flow control on a register FIS.

Transition HTCR1:1: When the entire FIS has been passed to the Link layer, the Transport layer shall indicate to the Link layer that the FIS transmit is complete and make a transition to the HTCR2: HT_TransStatus state.

Transition HTCR1:2: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1: HT_HostIdle state.

Transition HTCR1:3: When the Link layer indicates that an FIS is being received, the Transport Layer shall make a transition to the HTI1:HT_HostIdle state.

HTCR2: HT_TransStatus state: This state is when the entire FIS has been passed to the Link layer.

When in this state, the Transport layer shall wait for the Link and Phy ending status for the FIS and take appropriate error handling action if required.

Transition HTCR2:1: When the FIS status has been handled and no errors have been detected, the Transport layer shall transition to the HTI1: HT_HostIdle state.

Transition HTCR2:2: When the FIS status has been handled and at least one error has been detected, the Transport layer shall transition to the HTCR1: HT_Cntrl_FIS.

8.6.4 Host Transport transmit First-party DMA Setup – Device to Host or Host to Device FIS state diagram

This protocol transmits a DMA Setup – Device to Host or Host to Device FIS to a receiver. This FIS is a request by a transmitter for the receiver to program its DMA controller for a First-party DMA transfer and is followed by one or more Data FIS's that transfer data. The DMA Setup – Device to Host or Host to Device FIS request includes the transfer direction indicator, the host buffer identifier, the host buffer offset, the byte count, and the interrupt flag. The Transport layer shall not perform flow control on the DMA Setup – Host to Device or Device to Host FIS.

HTDMASTUP0: HT_DMASTUPFIS	Construct the DMA Setup – Host to Device or Device to Host FIS from the content provided by the Application layer and notifies Link to transfer.

1. FIS transfer complete	→	HT_TransStatus
2. Notification of illegal transition error received from Link layer	→	HT_HostIdle
3. Frame receipt indicated by Link layer	→	HT_HostIdle

HTPDMASTUP1: HT_TransStatus	Check Link and Phy transmission results and if an error occurred take appropriate action.

1. Status checked, and no error detected.	→	HT_HostIdle
2. Status checked, error detected, and no SRST.	→	HT_DMASTUPFIS
3. Status checked, error detected, and SRST.	→	HT_HostIdle

HTDMASTUP0: HT_DMASTUPFIS state: This state is entered when the Application requests the transmission of a DMA Setup – Host to Device or Device to Host FIS.

When in this state, the Transport layer shall construct a DMA Setup – Host to Device or Device to Host FIS, notify the Link layer that the FIS is to be transmitted, and pass the FIS to the Link layer. The Transport layer shall not perform flow control on a DMA Setup – Host to Device or Device to Host FIS.

Transition HTDMASTUP0:1: When the entire FIS has been passed to the Link layer, the Transport layer shall indicate to the Link layer that the FIS transmission is complete and make a transition to the HTDMASTUP1: HT_TransStatus state.

Transition HTDMASTUP0:2: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1: HT_HostIdle state.

Transition HTDMASTUP0:3: When the Link layer indicates that an FIS is being received, the Transport Layer shall make a transition to the HTI1:HT_HostIdle state.

HTPDMASTUP1: HT_TransStatus state: This state is entered when the entire FIS has been passed to the Link layer.

When in this state, the Transport layer shall wait for the Link and Phy ending status for the FIS and take appropriate error handling action if required.

Transition HTDMASTUP1:1: When the FIS status has been handled, and no error detected, the Transport layer shall transition to the HTI1: HT_HostIdle state.

Transition HTDMASTUP1:2: When the FIS status has been handled, an error detected and the host has not asserted the SRST by writing to the Device Control register, the Transport layer shall report status to the Link layer, and retry this transfer by transitioning to the HTDMASTUP0: HT_DMASTUPFIS state.

Transition HTDMASTUP1:3: When the host has asserted the SRST bit by writing to the Device Control register, or the ATAPI Device Reset command is issued, the Transport layer shall inform the Link layer to send SYNC primitives, and the Transport layer shall transition to the HTI1: HT_HostIdle state.

8.6.5 Host Transport transmit BIST Activate FIS

This protocol builds a BIST Activate FIS that tells the device to prepare to enter the appropriate Built-in Self-test mode. After successful transmission, the host Transport layer enters the idle state. The application layer, upon detecting successful transmission to the device shall then cause the host's Transport layer, Link layer and Physical layer to enter the appropriate mode for the transmission of the Built-in Test data defined by the FIS. The means by which the Transport, Link and Physical layers are placed into self-test mode are not defined by this specification.

HTXBIST1: HT_XmitBIST	Construct the BIST Activate FIS from the content provided by the Application layer and notify Link to transfer.	
1. FIS transfer complete	→	HT_TransBISTStatus
2. Notification of illegal transition error received from Link layer	→	HT_HostIdle
3. Frame receipt indicated by Link layer	→	HT_HostIdle

HTXBIST2: HT_TransBISTStatus	Check Link and Phy transmission results and if an error occurred take appropriate action.	
1. Status checked and no errors detected	→	HT_HostIdle
2. Status check and at least one error detected	→	HT_XmitBIST

HTXBIST1: HT_XmitBIST state: This state is entered to send a BIST FIS to the device.

Transition HTXBIST1:1: When the entire FIS has been passed to the Link layer, the Transport layer shall indicate to the Link layer that the FIS transmission is complete and make a transition to the HTXBIST2:HT_TransBISTStatus state.

Transition HTXBIST1:2: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1: HT_HostIdle state.

Transition HTXBIST1:3: When the Link layer indicates that an FIS is being received, the Transport Layer shall make a transition to the HTI1:HT_HostIdle state.

HTXBIST2: HT_TransBISTStatus state: This state is entered when the entire FIS has been passed to the Link layer.

Transition HTXBIST2:1: When the FIS transmission is completed and no errors have been detected the Transport layer shall transition to the HTI1:HT_HostIdle state.

Transition HTXBIST2:2: When the FIS transmission is completed and at least one error is detected the Transport layer shall transition to the HTXBIST1: HT_XmitBIST state.

8.6.6 Host Transport decompose Register FIS diagram

This protocol receives an FIS from the device containing new shadow register content and places that content into the shadow registers.

HTR1: HT_RegFIS	Place FIS contents from device into appropriate holding registers.		
	1. FIS transfer complete	→	HT_TransStatus
	2. Notification of illegal transition error received from Link layer	→	HT_HostIdle

HTR2: HT_TransStatus	Check Link and Phy transmission results and if an error occurred take appropriate action.		
	1. Status checked	→	HT_HostIdle

HTR1: HT_RegFIS state: This state is entered the Link layer has indicated that an FIS is being received and that the FIS is of the register type.

When in this state, the Transport layer shall decompose the register FIS and place the contents into the appropriate holding registers. The Transport layer shall not perform flow control on a register FIS.

Transition HTR1:1: When the entire FIS has been placed into the holding registers, the Transport layer shall make a transition to the HTR2: HT_TRANS_STATUS state.

Transition HTR1:2: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1: HT_HostIdle state.

HTR2: HT_TransStatus state: This state is when the entire FIS has been placed into the holding registers.

When in this state, the Transport layer shall wait for the Link and Phy ending status for the FIS and take appropriate error handling action if required.

Transition HTR2:1: When the FIS status has been handled and no errors detected, the contents of the holding registers shall be placed in the shadow registers and if the interrupt bit is set, the Transport layer shall set the interrupt pending flag. The Transport layer shall transition to the HTI1: HT_HostIdle state. When the FIS status has been handled and at least one error detected, the contents of the holding registers shall not be transferred to the shadow registers, error status shall be returned to the device, and the Transport layer shall transition to the HTI1: HT_HostIdle state.

8.6.7 Host Transport decompose a Set Device Bits FIS state diagram

This protocol receives an FIS from the device containing new Error and Status shadow register content and places that content into the Error and Status Shadow registers.

HTDB0:HT_DB_FIS	Receive Set Device Bits FIS	
1. FIS status checked and no error detected	→	HT_Dev_Bits
2. FIS status checked and error detected.	→	HT_HostIdle

HTDB1:HT_Dev_Bits	Load Error register and bits of the Status register	
1. Register bits loaded	→	HT_HostIdle

HTDB0:HT_DB_FIS state: This state is entered when the Link layer has indicated that an FIS being received and that the FIS is a Set Device Bits type.

When in this state, the Transport layer shall wait for the FIS reception to complete and for Link and Phy ending status to be posted.

Transition HTDB0:1: When the FIS reception is complete with no errors detected, the Transport layer shall transition to the HTDB1:HT_Dev_Bits state.

Transition HTDB0:2: When the FIS reception is complete with errors detected, the Transport layer shall return error status to the device and transition to the HTI1:HT_HostIdle state.

HTDB1:HT_Dev_Bits state: This state is entered when a Set Device Bits FIS has been received with no errors.

When in this state, the data in the Error field of the received FIS shall be loaded into the host adapter's shadow Error register. The data in the Status-Hi field of the received FIS shall be loaded into bits 6, 5, and 4 of the shadow Status register. The data in the Status-Lo field of the received FIS shall be loaded into bits 2, 1, and 0 of the shadow Status register. Bit 7, BSY, and bit 3, DRQ, in the shadow Status register shall not be changed. If the I bit in the FIS is set to one and if both the BSY bit and the DRQ bit in the shadow Status register are cleared to zero, then the host adapter shall enter an interrupt pending state.

Transition HTDB1:1: The Transport layer shall transition to the HTI1:HT_HostIdle state.

8.6.8 Host Transport decompose a DMA Activate FIS diagram

This protocol receives an FIS that requests a legacy DMA data out transfer. The data transfer is from the host to the device and the DMA Activate FIS causes the host adapter to transmit the data in a subsequent Data FIS.

HTDA1: HT_DMA_FIS		
1. Status checked and no error detected.	→	HT_DMAOTrans1
2. Status checked and error detected	→	HT_HostIdle
3. Notification of illegal transition error received from Link layer	→	HT_HostIdle

HTDA2: HT_DMAOTrans1	DMA controller initialized?	
1. DMA controller not initialized.	→	HT_DMAOTrans1
2. DMA controller initialized.	→	HT_DMAOTrans2
3. Notification of software reset or device reset command from the host application	→	HT_HostIdle

HTDA3: HT_DMAOTrans2	Activate DMA controller	
1. Transfer not complete and <8KB transmitted	→	HT_DMAOTrans2
2. Transfer not complete and 8KB transmitted	→	HT_DMAEnd
3. Abort notification from Link layer	→	HT_DMAEnd
4. Transfer complete	→	HT_DMAEnd
5. Notification of illegal transition error received from Link layer	→	HT_HostIdle
6. Notification of software reset or device reset command from the host application	→	HT_HostIdle

HTDA4: HT_DMAEnd	Check DMA Controller completion	
1. DMA controller actions completed, no error detected	→	HT_HostIdle
2. DMA controller actions completed, and error detected.	→	HT_HostIdle
3. Abort notification from Link layer, no error detected	→	HT_HostIdle
4. Abort notification from Link layer, error detected.	→	HT_HostIdle

HTDA5: HT_DMAITrans	Activate DMA controller if initialized, receive Data FIS.	
1. Transfer not complete	→	HT_DMAITrans
2. SRST asserted, or device reset command issued	→	HT_HostIdle
3. Transfer complete	→	HT_DMAEnd
4. Notification of illegal transition error received from Link layer	→	HT_HostIdle

HTDA1: HT_DMA_FIS state: This state is entered when the Link layer has indicated that an FIS is being received and the Transport layer has determined that a DMA Activate FIS is being received.

When in this state, the Transport layer shall determine the direction of the DMA transfer being activated.

Transition HTDA1:1: The Transport layer shall make a transition to the HTDA2: HT_DMAOTrans1 state. This transition occurs if no error is detected.

Transition HTDA1:2: When an error is detected, status is conveyed to the Link layer and to the application layer. The Transport layer shall make a transition to the HTI1:HT_HostIdle state.

Transition HTDA1:3: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1: HT_HostIdle state.

HTDA2: HT_DMAOTrans1 state: This state is entered when it is determined that the DMA transfer that is being activated is a transfer from host to device..

When in this state, the Transport layer shall determine if the DMA controller has been initialized.

Transition HTDA2:1: When the DMA controller has not yet been initialized, the Transport layer shall transition to the HTDA2: HT_DMAOTrans1 state.

Transition HTDA2:2: When the DMA controller has been initialized, the Transport layer shall transition to the HTDA3: HT_DMAOTrans2 state.

Transition HTDA2:3: When the host has asserted the SRST bit by writing to the Device Control register, or the ATAPI Device Reset command is written to the Command register, the Transport layer shall inform the Link layer to send SYNC primitives, and the Transport layer shall transition to the HTI1:HT_HostIdle state.

HTDA3: HT_DMAOTrans2 state: This state is entered when the DMA controller has been initialized.

When in this state, the Transport layer shall activate the DMA controller and pass data to the Link layer.

Transition HTDA3:1: When the transfer is not complete and less than 8KB of payload data has been transmitted, the Transport layer shall transition to the HTDA3: HT_DMAOTrans2 state.

Transition HTDA3:2: When the transfer is not complete but 8KB of payload data has been transmitted, the Link layer shall be notified to close the current frame and the Transport layer shall deactivate the DMA engine and transition to the HTDA4: HT_DMAEnd state.

Transition HTDA3:3: When notified by the Link layer that the DMA Abort primitive was received, the Transport layer shall transition to the HTDA4: HT_DMAEnd state.

Transition HTDA3:4: When the requested DMA transfer is complete, the Transport layer shall transition to the HTDA4: HT_DMAEnd state.

Transition HTDA3:5: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1: HT_HostIdle state.

Transition HTDA3:6: When the host has asserted the SRST bit by writing to the Device Control register, or the ATAPI Device Reset command is written to the Command register, the Transport layer shall inform the Link layer to send SYNC primitives, and the Transport layer shall transition to the HTI1:HT_HostIdle state.

HTDA4: HT_DMAEnd state: This state is entered when the DMA data transfer is complete.

When in this state, the Transport layer shall ensure that the activities of the DMA controller have completed.

Transition HTDA4:1: When the DMA controller has completed its activities, whether it has exhausted its transfer count or has been deactivated as a result of reaching the 8KB data payload limit, the Transport layer shall transition to the HTI1: HT_HostIdle state. This transition occurs if no error is detected.

Transition HTDA4:2: When an error is detected, status shall be reported to the Link and application layers. The Transport layer shall transition to the HTI1:HT_HostIdle state.

Transition HTDA4:3: When notified by the Link layer that a DMA Abort primitive was received, the transfer shall be truncated, and the Link layer notified to append CRC and end the frame. When it is determined that the transfer is completed with no error, the Transport layer shall make a transition to the HTI1:HT_HostIdle state.

Transition HTDA4:4: When notified by the Link layer that a DMA Abort primitive was received, the transfer shall be truncated, and the Link layer notified to append CRC and end the frame. When it is determined that the transfer is completed with an error, the Transport layer shall report status to the host and make a transition to the HTI1:HT_HostIdle state.

HTDA5: HT_DMAITrans state: This state is entered when the Transport layer has determined that the DMA transfer being activated is from device to host.

When in this state, the Transport layer shall activate the DMA controller if the DMA controller is initialized. A data frame will be received from the device and a received data Dword shall be placed in the data FIFO.

When in this state, the Transport layer shall wait until the Link layer has begun to receive the DMA data frame and data is available to be read by the host.

Transition HTDA5:1: When the transfer is not complete, the Transport layer shall transition to the HTDA5: HT_DMAITrans state. This includes the condition where the host DMA engine has not yet been programmed and the transfer is therefore held up until the DMA engine is prepared to transfer the received data to the destination memory locations.

Transition HTDA5:2: When the SRST bit is asserted by the host writing the Device Control register, or a device reset command has been written to an ATAPI device, the Link layer shall be informed to send a SYNC primitive, and the Transport layer shall transition to the HTI1:HT_Idle state.

Transition HTDA5:3: When the requested DMA transfer is complete, the Transport layer shall transition to the HTDA4: HT_DMAEnd state.

Transition HTDA5:4: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1: HT_HostIdle state.

8.6.9 Host Transport decompose a PIO Setup FIS state diagram

This protocol receives a PIO Setup FIS that requests a PIO data transfer. If the direction is from host to device, the Transport layer transmits a Data FIS to the device containing the PIO data. If the direction of transfer is from device to host, the Transport layer receives a Data FIS from the device. The PIO data must be sent in a single Data FIS.

HTPS1: HT_PS_FIS	Determine the direction of the requested PIO transfer.		
	1. Transfer host to device, no error detected. D bit = 0.	→	HT_PIOOTrans1
	2. Transfer device to host, no error detected. D bit = 1.	→	HT_HostIdle
	3. Error detected.	→	HT_HostIdle
	4. Notification of illegal transition error received from Link layer	→	HT_HostIdle

HTPS2: HT_PIOOTrans1	Place initial register content received from FIS into shadow registers.		
	1. Unconditional	→	HT_ HostIdle

HTPS3: HT_PIOOTrans2	Wait for Link layer to indicate data transfer complete		
	1. Transfer not complete	→	HT_PIOOTrans2
	2. Transfer complete	→	HT_PIOEnd
	3. Abort notification from Link layer	→	HT_PIOEnd
	4. Notification of illegal transition error received from Link layer	→	HT_HostIdle
	5. Notification of software reset or device reset command from the host application	→	HT_HostIdle

HTPS4: HT_PIOEnd	Place ending register content from PIO REQ FIS into shadow registers.		
	1. No error detected.	→	HT_HostIdle
	2. Error detected	→	HT_HostIdle

HTPS5: HT_PIOITrans1	Wait until initial PIO data received in data frame.		
	1. Received PIO data available.	→	HT_PIOITrans2[1]
	2. SRST asserted, or device reset command issued	→	HT_HostIdle
	3. Notification of illegal transition error received from Link layer	→	HT_HostIdle
	NOTE: 1. When transitioning to the HT_PIOITrans2 state, the starting status and I bit value from the PIO Setup FIS shall be transferred to the status shadow register and interrupt signal shall then reflect the value of the I bit.		

HTPS6: HT_PIOITrans2	Wait for Link layer to indicate data transfer complete.		
	1. Transfer not complete	→	HT_PIOITrans2
	2. Transfer complete	→	HT_PIOEnd
	3. Abort notification from Link layer	→	HT_PIOEnd
	4. Notification of illegal transition error received from Link layer	→	HT_HostIdle

HTPS1: HT_PS_FIS state: This state is entered when the Link layer has indicated that an FIS is being received and that the Transport layer has determined a PIO Setup FIS is being received.

When in this state, the Transport layer shall determine the direction of the requested PIO transfer and indicate that the last FIS sent was a PIO Setup.

Transition HTPS1:1: When the direction of transfer requested is from host to device (D=0), the Transport layer shall make a transition to the HTPS2: HT_PIOOTrans1 state. This transition occurs if no error is detected.

Transition HTPS1:2: When the direction of transfer requested is from device to host (D=1), the Transport layer shall make a transition to the HTI1:HT_HostIdle state. This transition occurs if no error is detected.

Transition HTPS1:3: When an error is detected, status shall be reported to the Link layer. The Transport layer shall make a transition to the HTI1:HT_HostIdle state.

Transition HTPS1:4: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1:HT_HostIdle state.

HTPS2: HT_PIOOTrans1 state: This state is entered when the direction of the requested PIO data transfer is from host to device.

When in this state, the Transport layer shall place the FIS initial register content into the shadow registers, the FIS byte count, and set the interrupt pending flag if the FIS indicates to do so.

Transition HTPS2:1: When the FIS initial register content has been placed into the shadow registers, interrupt pending set if requested, and the Transport layer is ready to begin transmitting the requested PIO data FIS, the Transport layer shall make a transition to the HTI1:HT_HostIdle state.

HTPS3: HT_PIOOTrans2 state: This state is entered when PIO data is available in the PIO FIFO to be passed the Link layer.

When in this state, the Transport layer shall wait for the Link layer to indicate that all data has been transferred.

NOTE – Since the software driver or BIOS sees DRQ set and BSY cleared, it continues writing the Data register filling the PIO FIFO.

Transition HTPS3:1: When the transfer is not complete, the Transport layer shall transition to the HTPS3: HT_PIOOTrans2 state.

Transition HTPS3:2: When the byte count for this DRQ data block is reached, the Transport layer shall transition to the HTPS4: HT_PIOEnd state.

Transition HTPS3:3: When notified by the Link layer that the DMA Abort primitive was received, the Transport layer shall transition to the HTPS4: HT_PIOEnd state.

Transition HTPS3:4: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1:HT_HostIdle state.

Transition HTPS3:5: When the host has asserted the SRST bit by writing to the Device Control register, or the ATAPI Device Reset command is written to the Command register, the Transport layer shall inform the Link layer to send SYNC primitives, and the Transport layer shall transition to the HTI1:HT_HostIdle state.

HTPS4: HT_PIOEnd state: This state is entered when the PIO data transfer is complete.

When in this state, the Transport layer shall place the ending register content from the received PIO request FIS into the shadow registers.

Transition HTPS4:1: When the ending register content for the PIO request FIS has been placed into the shadow registers and there were no errors detected with the transfer, the Transport layer shall transition to the HTI1: HT_HostIdle state.

Transition HTPS4:2: When the ending register content from the previous PIO Setup FIS has been placed into the shadow registers, the Transport layer shall transition to the HTI1:HT_HostIdle state. For data in transfers, the Transport layer shall notify the Link layer of any error encountered during the transfer, and the error shall be reflected in the end of frame handshake. If the transfer was not the final transfer for the PIO data in command, the device shall reflect the error status by transmitting an appropriate Register FIS to the host. If the transfer was the final transfer for the associated PIO data in command, the error condition is not detectable. For data out transfers, errors detected by the device shall be reflected in the end of frame handshake. The device shall reflect the error status by transmitting an appropriate Register FIS to the host.

HTPS5: HT_PIOITrans1 state: This state is entered when the direction of the PIO data transfer is device to host.

When in this state, the Transport layer shall wait until the Link layer has begun to receive the PIO data frame and data is available to be read by the host.

Transition HTPS5:1: When data is available for the host to read in the shadow Data register, the Transport layer shall place the initial register content received in the PIO Setup frame into the shadow registers and transition to the HTPS6: HT_PIOITrans2 state.

Transition HTPS5:2: When the host has asserted the SRST bit by writing to the Device Control register, or the ATAPI Device Reset command is issued, the Transport layer shall inform the Link layer to send SYNC primitives, and the Transport layer shall transition to the HTI1: HT_HostIdle state.

Transition HTPS5:3: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1:HT_HostIdle state.

HTPS6: HT_PIOITrans2 state: This state is entered when PIO data is available in the PIO FIFO to be read by the host and the initial shadow register content has been set.

When in this state, the Transport layer shall wait for the Link layer to indicate that the data transfer is complete

Transition HTPS6:1: When the transfer is not complete, the Transport layer shall transition to the HTPS6: HT_PIOITrans2 state.

Transition HTPS6:2: When the byte count for this DRQ data block is reached, the Transport layer shall transition to the HTPS5: HT_PIOEnd state.

Transition HTPS6:3: When notified by the Link layer that the DMA Abort primitive was received, the Transport layer shall transition to the HTPS5: HT_PIOEnd state.

Transition HTPS6:4: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1:HT_HostIdle state.

8.6.10 Host Transport decompose a First-party DMA Setup FIS state diagram

This protocol receives an FIS that sets up the host adapter DMA controller to allow the transfer of a first party DMA data FIS to the host. The actual DMA Activate FIS will be transmitted as a separate DMA Activate FIS protocol following the completion of this protocol.

HTDS1: HT_DS_FIS	Initialize the DMA controller for a first party DMA with the content of the request FIS.	
1. No error detected.	→	HT_HostIdle
2. Error detected.	→	HT_HostIdle
3. Notification of illegal transition error received from Link layer	→	HT_HostIdle

HTDS1: HT_DR_FIS state: This state is entered when the Link layer has indicated that an FIS is being received and that the Transport layer has determined the FIS is of the first party DMA in request type.

When in this state, the Transport layer shall initialize the DMA controller with content from the FIS.

Transition HTDS1:1: When the DMA controller has been initialized, the Transport layer shall transition to the HTI1: HT_HostIdle state. This transition is made if no error is detected.

Transition HTDS1:2: When an error is detected, status shall be reported to the Link layer. The Transport layer shall transition to the HTI1:HI_HostIdle state.

Transition HTDS1:3: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1:HT_HostIdle state.

8.6.11 Host transport decompose a BIST Activate FIS state diagram

This protocol receives an FIS that instructs the host to enter one of several Built-in Self-test modes that cause the host to retransmit the data it receives. If the mode is supported the Host's application layer will place both the transmit and receive portions of the Transport, Link and/or Physical layers into appropriate state to perform the loopback operation.

HTRBIST1: HT_RcvBIST	Determine validity of loopback mode requested.	
1. Status checked, no error detected and Loopback mode valid	→	HT_BISTTrans1
2. Status checked, no error detected and Loopback mode is invalid or not supported.	→	HT_HostIdle
3. Status checked and error detected	→	HT_HostIdle
4. Notification of illegal transition error received from Link layer	→	HT_HostIdle

HTRBIST2: HT_BISTTrans1	Notify application layer of desired BIST modes	
1. Unconditional	→	HT_HostIdle

HTRBIST1: HT_RcvBIST state: This state is entered when the link layer has indicated that an FIS is being received and the Transport layer has determined that a BIST Activate FIS is being received.

When in this state, the Transport layer shall determine the validity of the loopback request.

Transition HTRBIST1:1: If no reception error is detected and the FIS contents indicate a form of loopback request that is supported by the host the Transport layer shall make a transition to the HTRBIST2: HT_BISTTrans1 state.

Transition HTRBIST1:2: If no reception error is detected and the FIS contents indicate a form of loopback request that is not supported by the host the Transport layer shall make a transition to the HTI1:HT_HostIdle state.

Transition HTRBIST1:3: If a reception error is indicated the Transport layer shall make a transition to the HTI1:HT_HostIdle state.

Transition HTRBIST1:4: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1:HT_HostIdle state.

HTRBIST2: HT_BISTTrans1 state: This state is entered when the Transport layer has determined that a valid BIST Activate FIS has been received.

Having received a valid FIS, the Transport layer informs the application layer that it should place the Transport, Link and Physical layers into the appropriate modes to loop the received data back to the transmitter. The method by which this is performed is not defined by this specification.

Transition HTRBIST2:1: When the Application layer has been notified the Transport layer shall transition to the HTI1:HostIdle state.

8.7 Device transport states

8.7.1 Device transport idle state diagram

DTI0: DT_DeviceIdle	Device waits for FIS or FIS request.		
	1. Transmit of RegDHFIS requested by Application layer	→	DT_RegDHFIS
	2. Transmission of Set Device Bits requested by Application	→	DT_DB_FIS
	3. Transmit of PIOSTUPFIS requested by Application layer	→	DT_PIOSTUPFIS
	4. Transmit of DMAACTFIS requested by Application layer	→	DT_DMAACTFIS
	5. Transmit of DMASTUPFIS requested by Application layer	→	DT_DMASTUPDHFIS
	6. Transmit of DATAFIS requested by Application layer	→	DT_DATAIFIS
	7. Transmit of BIST FIS requested by Application layer	→	DT_XmitBIST
	8. Frame receipt indicated by Link layer	→	DT_ChkTyp

DTI1: DT_ChkTyp	Received FIS type checked.		
	1. Register FIS type detected	→	DT_RegHDFIS
	2. Data FIS type detected	→	DT_DATAOFIS
	3. DMA Setup FIS type detected	→	DT_DMASTUPHDFIS
	4. BIST Activate FIS type detected	→	DT_rcvBIST
	5. Notification of illegal transition error received from Link layer or unrecognized FIS type	→	DT_DeviceIdle

DTI0: DT_DeviceIdle state: This state is entered when a Frame Information Structure (FIS) transaction has been completed by the Transport layer.

When in this state, the Transport layer waits for the Application layer to indicate that an FIS is to be transmitted or the Link layer to indicate that an FIS is being received.

Transition DTI0:1: When the Application layer indicates that a register FIS is to be transmitted, the Transport layer shall make a transition to the DTR0: DT_RegDHFIS state.

Transition DTI0:2: When the Application layer indicates that a Set Device Bits FIS is to be transmitted, the Transport layer shall make a transition to the DTDB0:DT_DB_FIS state.

Transition DTI0:3: When the Application layer indicates that a PIO Setup FIS is to be transmitted, the Transport layer shall make a transition to the DTPIOSTUP0: DT_PIOSTUPFIS state.

Transition DTI0:4: When the Application layer indicates that a DMA Activate FIS is to be transmitted, the Transport layer shall make a transition to the DTDMAACT0: DT_DMAACTFIS state.

Transition DTI0:5: When the Application layer indicates that a First-party DMA Setup FIS is to be transmitted, the Transport layer shall make a transition to the DTDMASTUP0: DT_DMASTUPDHFIS state.

Transition DTI0:6: When the Application layer indicates that a Data FIS is to be transmitted, the Transport layer shall make a transition to the DTDATAI0: DT_DATAIFIS state.

Transition DTI0:7: When the Application layer indicates that a BIST Activate FIS is to be transmitted, the Transport layer shall make a transition to the DTXBIST1:DT XmitBIST state.

Transition DTI0:8: When the Link layer indicates that an FIS is being received, the Transport layer shall make a transition to the DTI1: DT_ChkTyp state.

DTI1: DT_ChkTyp state: This state is entered when the Transport layer is idle and Link layer indicates that an FIS is being received.

When in this state, the Transport layer checks the FIS type of the incoming FIS.

Transition DTI1:1: When the incoming FIS is a Register – Host to Device FIS type, the Transport layer shall make a transition to the DTCMD0: DT_RegHDFIS state.

Transition DTI1:2: When the incoming FIS is a Data - Host to Device FIS type, the Transport layer shall make a transition to the DTDATAO0: DT_DATAOFIS state.

Transition DTI1:3: When the incoming FIS is a DMA Setup FIS type, the Transport layer shall make a transition to the DTSTP0: DT_ DMASTUPHDFIS state.

Transition DTI1:4: When the incoming FIS is a BIST Activate FIS type, the Transport layer shall make a transition to the DTRBIST1:DT Rcv BIST state.

Transition DTI1:5: When the Transport layer receives notification from the Link layer of an illegal state transition or the FIS type is not recognized, the Transport layer shall make a transition to the DTI0: DT_DeviceIdle state.

8.7.2 Device Transport send Register – Device to Host state diagram

This protocol builds a Register – Device to Host FIS that contains the register content and sends it to the host when the Application layer requests the transmission.

DTR0: DT_RegDHFIS	Construct Register – Device to Host FIS from the content of the registers and notify Link to transfer.		
	1. FIS transfer complete	→	DT_TransStatus
	2. Notification of illegal transition error received from Link layer	→	DT_DeviceIdle

DTR1: DT_TransStatus	Check Link and Phy transmission results and if an error occurred take appropriate action.		
	1. Status checked, and no error detected.	→	DT_DeviceIdle
	2. Status checked, and error detected.	→	DT_RegDHFIS

DTR0: DT_RegDHFIS state: This state is entered the Application requests the transmission of a Register – Device to Host FIS.

When in this state, the Transport layer shall construct a Register – Device to Host FIS, notify the Link layer that the FIS is to be transmitted, and pass the FIS to the Link layer. The Transport layer shall not perform flow control on the FIS transmission.

Transition DTR0:1: When the entire FIS has been passed to the Link layer, the Transport layer shall indicate to the Link layer that the FIS transmission is complete and make a transition to the DTR1: DT_TransStatus state.

Transition DTR0:2: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the DTI0: DT_DeviceIdle state.

DTR1: DT_TransStatus state: This state is when the entire FIS has been passed to the Link layer.

When in this state, the Transport layer shall wait for the Link and Phy ending status for the FIS and take appropriate error handling action if required.

Transition DTR1:1: When the FIS status has been handled, and no error detected, the Transport layer shall transition to the DTI0: DT_DeviceIdle state.

Transition DTR1:2: When the FIS status has been handled, and an error detected, the Transport layer shall report status to the Link layer, and retry this transfer by transitioning to the DT_RegDHFISstate.

8.7.3 Device Transport send Set Device Bits FIS state diagram

This protocol sends a Set Device Bits FIS to the host adapter when the Application layer requests the transmission.

DTDB0: DT_DB_FIS	Inform Link to transmit Set Device Bits FIS		
	1. FIS transfer complete	→	DT_TransStatus

DTDB1: DT_TransStatus	Check Link and Phy transmission results and if an error occurred take appropriate action.		
	1. Status checked and no error detected	→	DT_DeviceIdle
	2. Status checked and error detected.	→	DT_DB_FIS

DTDB0:DT_DB_FIS state: This state is entered when the Application layer requests the transmission of a Set Device Bits FIS.

When in this state, the Transport layer shall construct a Set Device Bits FIS, notify the Link layer that the FIS is to be transmitted, and pass the FIS to the Link layer. The Transport layer shall not perform flow control on the FIS transmission.

Transition DTDB0:1: When the entire FIS has been passed to the Link layer, the Transport layer shall indicate to the Link layer that the FIS transmission is complete and make a transition to the DTDB1:DT_TransStatus state.

DTDB1:DT_TransStatus state: This state is entered when the entire FIS has been passed to the Link layer.

When in this state, the Transport layer shall wait for the Link and Phy ending status for the FIS and take appropriate error handling action if required.

Transition DTDB1:1: When the FIS status has been handled and no error detected, the Transport layer shall transition to the DTI0:DT_DeviceIdle state.

Transition DTDB1:2: When the FIS status has been handled and an error detected, the Transport layer shall report status to the Link layer and retry this transfer by transitioning to the DTDB0:DT_DB_FIS state.

8.7.4 Device Transport transmit PIO Setup – Device to Host FIS state diagram

This protocol transmits a PIO Setup – Device to Host FIS to the host. Following this PIO Setup frame, a single data frame containing PIO data shall be transmitted or received depending on the state of the D bit in the PIO Setup frame. The Transport layer shall not perform flow control on the PIO Setup FIS.

DTPIOSTUP0: DT_PIOSTUPFIS	Construct PIO Setup – Device to Host FIS from the content provided by the Application layer and notify Link to transfer. This FIS shall include the beginning and ending register content, the byte count, and the interrupt flag.		
1. FIS transfer complete		→	DT_TransStatus
2. Notification of illegal transition error received from Link layer		→	DT_DeviceIdle

DTPIOSTUP1: DT_TransStatus	Check Link and Phy transmission results and if an error occurred take appropriate action.		
1. Status checked, and no error detected.		→	DT_Device Idle
2. Status checked, and error detected.		→	DT_PIOSTUPFIS

DTPIOSTUP0: DT_PIOSTUPFIS state: This state is entered the Application layer requests the transmission of a PIO Setup – Device to Host FIS.

When in this state, the Transport layer shall construct a PIO Setup – Device to Host FIS, notify the Link layer that the FIS is to be transmitted, and pass the FIS to the Link layer. The Transport layer shall not perform flow control on a register FIS.

Transition DTPIOSTUP0:1: When the entire FIS has been passed to the Link layer, the Transport layer shall indicate to the Link layer that the FIS transmission is complete and make a transition to the DTPIOSTUP1: DT_TransStatus state.

Transition DTPIOSTUP0:2: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the DTI0: DT_DeviceIdle state.

DTPIOSTUP1: DT_TransStatus state: This state is when the entire FIS has been passed to the Link layer.

When in this state, the Transport layer shall wait for the Link and Phy ending status for the FIS and take appropriate error handling action if required.

Transition DTPIOSTUP1:1: When the FIS status has been handled and no error detected, the Transport layer shall transition to the DTI0: DT_DeviceIdle state.

Transition DTPIOSTUP1:**2**: When the FIS status has been handled and an error detected, the Transport layer shall report status to the Link layer and retry this transfer by transitioning to the DT_PIOSTUPFIS state.

8.7.5 Device Transport transmit Legacy DMA Activate FIS state diagram

This protocol transmits a Legacy DMA Activate FIS to the host adapter. Following this Legacy DMA Activate frame, a data frame of DMA data shall be sent from the host to the device. The Transport layer shall not perform flow control on the DMA Activate FIS.

DTDMAACT0: DT_DMAACTFIS	Construct Legacy DMA Activate FIS from the content provided by the Application layer and notify Link to transfer.	
1. FIS transfer complete	→	DT_TransStatus
2. Notification of illegal transition error received from Link layer	→	DT_DeviceIdle

DTDMAACT1: DT_TransStatus	Check Link and Phy transmission results and if an error occurred take appropriate action.	
1. Status checked, and no error detected.	→	DT_DeviceIdle
2. Status checked, and error detected.	→	DT_DMAACTFIS

DTDMAACT0: DT_DMAACTFIS state: This state is entered the Application requests the transmission of a Legacy DMA Activate FIS.

When in this state, the Transport layer shall construct a Legacy DMA Activate FIS, notify the Link layer that the FIS is to be transmitted, and pass the FIS to the Link layer. The Transport layer shall not perform flow control on a register FIS.

Transition DTDMAACT0:1: When the entire FIS has been passed to the Link layer, the Transport layer shall indicate to the Link layer that the FIS transmission is complete and make a transition to the DTDMAACT1: DT_TransStatus state.

Transition DTDMAACT0:2: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the DTI0: DT_DeviceIdle state.

DTDMAACT1: DT_TransStatus state: This state entered is when the entire FIS has been passed to the Link layer.

When in this state, the Transport layer shall wait for the Link and Phy ending status for the FIS and take appropriate error handling action if required.

Transition DTDMAACT1:1: When the FIS status has been handled and no error detected, the Transport layer shall transition to the DTI0: DT_DeviceIdle state.

Transition DTDMAACT1:**2**: When the FIS status has been handled and an error detected, the Transport layer shall report status to the Link layer and retry this transfer by transitioning to the DT_DMAACTFIS state.

8.7.6 Device Transport transmit First-party DMA Setup – Device to Host FIS state diagram

This protocol transmits a First-party DMA Setup – Device to Host FIS to the host adapter. This FIS is a request by the device for the host adapter to program the DMA controller for a First-party DMA transfer and is followed by one or more Data FIS's that transfer the data to or from the host adapter depending on the direction of the transfer. The First-party DMA Setup – Device to Host request includes the transfer direction indicator, the host buffer identifier, the host buffer offset, the byte count, and the interrupt flag. The Transport layer shall not perform flow control on the First-party DMA Setup – Device to Host FIS.

DTDMASTUP0: DT_DMASTUPDHFIS	Construct the First-party DMA Setup – Device to Host FIS from the content provided by the Application layer and notify Link to transfer.	
1. FIS transfer complete	→	DT_TransStatus
2. Notification of illegal transition error received from Link layer	→	DT_DeviceIdle

DTDMASTUP1: DT_TransStatus	Check Link and Phy transmission results and if an error occurred take appropriate action.	
1. Status checked, and no error detected.	→	DT_DeviceIdle
2. Status checked, and error detected.	→	DT_DMASTUPFIS

DTDMASTUP0: DT_DMASTUPDHFIS state: This state is entered when the Application requests the transmission of a First-party DMA Setup – Device to Host FIS.

When in this state, the Transport layer shall construct a First-party DMA Setup – Device to Host FIS, notify the Link layer that the FIS is to be transmitted, and pass the FIS to the Link layer. The Transport layer shall not perform flow control on a First-party DMA Setup – Device to Host FIS.

Transition DTDMASTUP0:1: When the entire FIS has been passed to the Link layer, the Transport layer shall indicate to the Link layer that the FIS transmission is complete and make a transition to the DTDMASTUP1: DT_TransStatus state.

Transition DTIDMASTUP0:2: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the DTI0: DT_DeviceIdle state.

DTPDMASTUP1: DT_TransStatus state: This state is entered when the entire FIS has been passed to the Link layer.

When in this state, the Transport layer shall wait for the Link and Phy ending status for the FIS and take appropriate error handling action if required.

Transition DTDMASTUP1:1: When the FIS status has been handled and no error detected, the Transport layer shall transition to the DTI0: DT_DeviceIdle state.

Transition DTDMASTUP1:2: When the FIS status has been handled and an error detected, the Transport layer shall report status to the Link layer and retry this transfer by transitioning to the DT_DMASTUPFIS state.

8.7.7 Device Transport transmit Data – Device to Host FIS diagram

This protocol builds a Data – Device to Host FIS.

DTDATAI0: DT_DATAIFIS	Construct Data – Device to Host FIS content from the content provided by the Application layer and notify Link to transfer.	
1. Unconditional	→	DT_DATAITrans

DTDATAI1: DT_DATAITrans	Pass data Dwords from data FIFO to Link layer	
1. Transfer not complete	→	DT_DATAITrans
2. Transfer complete	→	DT_DATAIEnd
3. Application layer requests termination of DMA in transfer.	→	DT_DATAIEnd
4. Notification of illegal transition error received from Link layer	→	DT_DeviceIdle

DTDATAI2: DT_DATAIEnd	Check Link and Phy transmission results and if an error occurred take appropriate action.	
1. Status checked, and no error detected.	→	DT_DeviceIdle
2. Status checked, and error detected.	→	DT_DeviceIdle
3. Application layer requests termination of DMA in transfer, no error detected.	→	DT_DeviceIdle
4. Application layer requests termination of DMA in transfer, error detected.	→	DT_DeviceIdle
5. Notification of illegal transition error received from Link layer	→	DT_DeviceIdle

DTDATAI0: DT_DATAFIS state: This state is entered when the Application layer has requested the transmission of a Data – Device to Host FIS.

When in this state the Transport layer shall pass a portion of the DMA data to the Link layer.

Transition DTDATAI0:1: When ready and there is data in the FIFO to be passed to the Link layer, the Transport layer shall transition to the DTDATAI1: DT_DATAITrans state.

DTDATAI1: DT_DATAITrans state: This state is entered when data is available in the FIFO to be passed the Link layer.

When in this state, the Transport layer shall pass a Dword of data from the FIFO to the Link layer.

Transition DTDATAI1:1: When the transfer is not complete, the Transport layer shall transition to the DTDATAI1: DT_DATAITrans state.

Transition DTDATAI1:2: When the transfer is complete, the Transport layer shall transition to the DTDATAI2: DT_DATAIEnd state.

Transition DTDATAI1:3: When the Application layer requests that a DMA operation is to be aborted, the Transport Layer shall transition to the DTDATAI2:DT_DATAIEnd state.

Transition DTDATAI1:4: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the DTI0: DT_DeviceIdle state.

DTDATAI2: DT_DATAIEnd state: This state is entered when the data transfer is complete or an abort has been requested by the application layer.

When in this state, the Transport layer shall wait for the Link and Phy ending status for the FIS and take appropriate error handling action if required.

Transition DTDATAI2:1: When the FIS status has been handled and no error detected, the Transport layer shall transition to the DTI0: DT_DeviceIdle state.

Transition DTDATAI2:2: When the FIS status has been handled and an error detected, status shall be reported to the Link layer. The Transport layer shall transition to the DTI0: DT_DeviceIdle state.

Transition DTDATAI2:3: When the application layer requests the termination of a DMA data in transaction, it will report the abort condition to the Link, wait for an EOF and, if no error is detected, make a transition to the DTI0:DT_DeviceIdle state.

Transition DTDATAI2:4: When the application layer requests the termination of a DMA data in transaction, it will report the abort condition to the Link, wait for an EOF, and if an error is detected, report the error to the Link layer, and make a transition to the DTI0:DT_DeviceIdle state.

Transition DTDATAI2:5: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the DTI0: DT_DeviceIdle state.

8.7.8 Device Transport transmit BIST Activate FIS diagram

This protocol builds a BIST Activate FIS that tells the host to prepare to enter the appropriate Built-in Self Test mode. After successful transmission, the device Transport layer enters the idle state. The application layer, upon detecting successful transmission to the host shall then cause the device's Transport layer, Link layer and Physical layer to enter the appropriate mode for the transmission of the Built-in Test data defined by the FIS. The means by which the Transport, Link and Physical layers are placed into self-test mode are not defined by this specification.

DTXBIST1: DT_XmitBIST	Construct the BIST Activate FIS from the content provided by the Application layer and notifies Link to transfer.		
	1. FIS transfer complete	→	DT_TransBISTStatus
	2. Notification of illegal transition error received from Link layer	→	DT_DeviceIdle

DTXBIST2: DT_TransBISTStatus	Check Link and Phy transmission results and if an error occurred take appropriate action.		
	1. Status checked and no errors detected	→	DT_DeviceIdle
	2. Status check and at least one error detected	→	DT_XmitBIST

DTXBIST1: DT_XmitBIST state: This state is entered to send a BIST FIS to the host.

Transition DTXBIST1:1: When the entire FIS has been passed to the Link layer, the Transport layer shall indicate to the Link layer that the FIS transmission is complete and make a transition to the DTXBIST2:DT_TransBISTSTatus state.

Transition DTXBIST1:2: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the DTI0: DT_DeviceIdle state.

DTXBIST2: DT_TransBISTStatus state: This state is entered when the entire FIS has been passed to the Link layer.

Transition DTXBIST2:1: When the FIS transmission is completed and no errors have been detected, the Transport layer shall transition to the DTI0: DT_DeviceIdle state.

Transition DTXBIST2:2: When the FIS transmission is completed and at least one error is detected, the Transport layer shall transition to the DTXBIST1: DT_XmitBIST state.

8.7.9 Device Transport decompose Register – Host to Device state diagram

This protocol receives a Register – Host to Device FIS, places received register content into the device registers, and notifies the Application layer of the FIS receipt. The Transport layer shall not flow control the receipt of this FIS.

DTCMD0: DT_RegHDFIS	Receive a Register – Host to Device FIS	
1. FIS transfer complete, and no error detected.	→	DT_DeviceIdle
2. FIS transfer complete, and error detected	→	DT_DeviceIdle
3. Notification of illegal transition error received from Link layer	→	DT_DeviceIdle

DTCMD0: DT_RegHDFIS state: This state is entered when the receipt of a DT_RegHDFISFIS is recognized.

When in this state, the Transport layer shall receive the FIS and place the contents of the FIS into the device registers when it is determined that the FIS was received without error. The Transport layer shall not perform flow control on a command FIS.

Transition DTCMD0:1: When the entire FIS has been received from the Link layer without error, the Transport layer shall indicate to the Application layer that a command FIS was received and make a transition to the DTI0: DT_DeviceIdle state.

Transition DTCMD0:2: When the entire FIS has been received from the Link layer and an error has been detected, status shall be sent to the Link layer. The Transport layer shall make a transition to the DTI0:DT_DeviceIdle state.

Transition DTCMD0:3: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the DTI0: DT_DeviceIdle state.

8.7.10 Device Transport decompose Data (Host to Device) FIS state diagram

This protocol receives a Data - Host to Device FIS.

DTDATAO0: DT_DATAOFIS	Prepare to receive data		
	Unconditional	→	DT_DATAOREC

DTDATAO1: DT_DATAOREC	Place received data Dword into data FIFO and signal Link to continue transfer.		
	1. Transfer not complete	→	DT_DATAOREC
	2. Transfer complete	→	DT_DeviceIdle
	3. Abort Transfer from Application layer	→	DT_DeviceAbort
	4. Notification of illegal transition error received from Link layer	→	DT_DeviceIdle

DTDATAO2: DT_DeviceAbort	Signal Link to abort transfer.		
	1. Transfer not complete	→	DT_DATAOREC
	2. Transfer complete	→	DT_ DeviceIdle

DTDATAO0: DT_DATAOFIS state: This state is entered when the Link layer has indicated that an FIS is being received and that the Transport layer has determined the FIS is of Data - Host to Device type.

When in this state, the Transport layer shall prepare to receive the data.

Transition DTDATAO0:1: When ready to receive the data, the Transport layer shall make a transition to the DTDATAO1: DT_DATAOREC state.

DTDATAO1: DT_DATAOREC state: This state is entered when the Transport layer is ready to receive the data.

When in this state, the Transport layer shall wait for the Link layer to indicate the transfer is complete.

Transition DTDATAO1:1: When Link layer has not indicated that the end of the FIS has been reached, the Transport layer shall transition to the DTDATAO1: DT_DATAOREC state.

Transition DTDATAO1:2: When the Link layer indicates that the end of the FIS has been reached, the Transport layer shall transition to the DTI0: DT_DeviceIdle state.

Transition DTDATAO1:3: When the Application layer indicates that the FIS is to be aborted, the Transport layer shall transition to the DTDATAO2: DT_DeviceAbort state.

Transition DTDATAO1:4: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the DTI0: DT_DeviceIdle state.

DTDATAO2: DT_DeviceAbort state: This state is entered when the application layer indicates that the current transfer is to be aborted.

When in this state, the Transport layer shall signal the Link layer to Abort the incoming transmission and return to either DT_DATAOREC or DT_DATAOhold, depending upon the current state of the

FIFO. If the abort occurs coincident with an end of transfer indication from the Link, then the transition to DTI0: DT_DeviceIdle is also accommodated. After issuing an Abort, the Transport returns to normal data transfer, and awaits the end of transfer indication from the Link.

Transition DTDATAO2:1: Inform Link layer to issue an abort. When Transfer is not complete, the Transport layer shall transition to the DTDATAO1: DT_DATAOREC state.

Transition DTDATAO2:2: Inform Link layer to issue an abort. When the Link layer indicates that the end of the FIS has been reached, the Transport layer shall transition to the DTI0: DT_DeviceIdle state.

8.7.11 Device Transport decompose DMA Setup – Host to Device or Device to Host state diagram

This protocol receives a DMA Setup – Host to Device or Device to Host FIS, passes received DMA Setup content, and notification of FIS receipt to the Application layer. The Transport layer shall not flow control the receipt of this FIS.

DTSTP0: DT_DMASTUPHDFIS	Receive a Register – Host to Device FIS	
1. FIS transfer complete, and no error detected.	→	DT_DeviceIdle
2. FIS transfer complete, and error detected	→	DT_DeviceIdle
3. Notification of illegal transition error received from Link layer	→	DT_DeviceIdle

DTSTP0: DT_DMASTUPHDFIS state: This state is entered when the receipt of a DT_DMASTUP FIS is recognized.

Transition DTSTP0:1: When the entire FIS has been received from the Link layer without error, the Transport layer shall indicate to the Application layer that a command FIS was received and make a transition to the DTI0: DT_DeviceIdle state.

Transition DTSTP0:2: When the entire FIS has been received from the Link layer and an error has been detected, status shall be sent to the link layer. The Transport layer shall make a transition to the DTI0: DT_DeviceIdle state.

Transition DTSTP0:3: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the DTI0: DT_DeviceIdle state.

8.7.12 Device Transport decompose a BIST Activate FIS state diagram

This protocol receives an FIS that instructs the device to enter one of several Built-in Self-test modes that cause the device to retransmit the data it receives. If the mode is supported the Device's application layer will place both the transmit and receive portions of the Transport, Link and/or Physical layers into appropriate state to perform the loopback operation.

DTRBIST1: DT_RcvBIST	Determine validity of loopback mode requested.	
1. Status checked, no error detected and Loopback mode valid	→	DT_BISTTrans1
2. Status checked, no error detected and Loopback mode is invalid or not supported.	→	DT_DeviceIdle
3. Status checked and error detected	→	DT_DeviceIdle
4. Notification of illegal transition error received from Link layer	→	DT_DeviceIdle

DTRBIST2: DT_BISTTrans1	Notify application layer of desired BIST modes	
1. Unconditional	→	DT_DeviceIdle

DTRBIST1: DT_RcvBIST state: This state is entered when the Link layer has indicated that an FIS is being received and the Transport layer has determined that a BIST Activate FIS is being received.

When in this state, the Transport layer shall determine the validity of the loopback request.

Transition DTRBIST1:1: If no reception error is detected and the FIS contents indicate a form of loopback request that is supported by the host the Transport layer shall make a transition to the DTRBIST2: DT_BISTTrans1 state.

Transition DTRBIST1:2: If no reception error is detected and the FIS contents indicate a form of loopback request that is not supported by the host the Transport layer shall make a transition to the DTI0:DT_DeviceIdle state.

Transition DTRBIST1:3: If a reception error is indicated the Transport layer shall make a transition to the DTI0:DT_DeviceIdle state.

Transition DTRBIST1:4: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the DTI0: DT_DeviceIdle state.

DTRBIST2: DT_BISTTrans1 state: This state is entered when the Transport layer has determined that a valid BIST Activate FIS has been received.

Having received a valid FIS, the Transport layer informs the application layer that it should place the Transport, Link and Physical layers into the appropriate modes to loop the received data back to the transmitter. The method by which this is performed is not defined by this specification.

Transition DTRBIST2:1: When the Application layer has been notified the Transport layer shall transition to the DTI0:DT_DeviceIdle state.

9 Device command layer protocol

9.1 Power-on and COMRESET protocol diagram

If the host asserts Hard Reset regardless of the power management mode or the current device command layer state, the device shall execute the hardware reset protocol.

DHR0: Hardware_reset_asserted	Wait.	
1. COMRESET negated.	→	DHR1: Execute_diagnostics

DHR1: Execute_diagnostics	Initialize hardware and execute diagnostics.	
1. Initialization and diagnostics completed successfully.	→	DHR2: Send_good_status
2. Initialization completed and diagnostics failed.	→	DHR3: Send_bad_status

DHR2: Send_good_status	Request transmission of Register FIS with good status.	
1. Register FIS transmitted.	→	DI0: Device_idle

DHR3: Send_bad_status	Request transmission of Register FIS with bad status.	
1. Register FIS transmitted.	→	DI0: Device_idle

DHR0: Hardware_reset_asserted: This state is entered when the Transport layer indicates that the COMRESET signal is asserted.

When in this state, the device awaits the negation of the COMRESET signal.

Transition DHR0:1: When the Transport layer indicates that the COMRESET signal has been negated, the device shall transition to the DHR1: Execute_diagnostics state.

DHR1: Execute_diagnostics: This state is entered when the Transport layer indicates that the COMRESET signal has been negated.

When in this state, the device initializes the device hardware and executes its power-up diagnostics.

Transition DHR1:1: When the device hardware has been initialized and the power-up diagnostics successfully completed, the device shall transition to the DHR2: Send_good_status state.

Transition DHR1:2: When the device hardware has been initialized and the power-up diagnostics failed, the device shall transition to the DHR3: Send_bad_status state.

DHR2: Send_good_status: This state is entered when the device hardware has been initialized and the power-up diagnostics successfully completed.

When in this state, the device requests that the Transport layer transmit a Register FIS to the host. If the device does not implement the PACKET command feature set the register content shall be:

Sector Count	01h
Sector Number	01h
Cylinder Low	00h
Cylinder High	00h
Device/Head	00h
Error	01h
Status	00h-70h (see note)
NOTE – Setting of bit 6 thru 4 in the Status register are device specific.	

If the device implements the PACKET command feature set, the register content shall be:

Sector Count	01h
Sector Number	01h
Cylinder Low	14h
Cylinder High	EBh
Device/Head	00h
Error	01h
Status	00h

Transition DHR2:1: When the Transport layer indicates that the Register FIS has been transmitted, the device shall transition to the DI0: Device_Idle state.

DHR3: Send_bad_status: This state is entered when the device hardware has been initialized and the power-up diagnostics failed.

When in this state, the device requests that the Transport layer transmit a Register FIS to the host. If the device does not implement the PACKET command feature set the register content shall be:

Sector Count	01h
Sector Number	01h
Cylinder Low	00h
Cylinder High	00h
Device/Head	00h
Error	00h, 02h-7Fh
Status	00h-70h (see note)
NOTE – Setting of bit 6 thru 4 in the Status register are device specific.	

If the device implements the PACKET command feature set, the register content shall be:

Sector Count	01h
Sector Number	01h
Cylinder Low	14h
Cylinder High	EBh
Device/Head	00h
Error	00h, 02h-7Fh
Status	00h

Transition DHR3:1: When the Transport layer indicates that the Register FIS has been transmitted, the device shall transition to the DI0: Device_Idle state.

9.2 Device Idle protocol

The state diagram below describes the idle protocol for a device. States and transitions preceded by an * are utilized only when queuing is implemented.

DI0: Device_idle	Wait.	
1. FIS receipt	→	DI1: Check_FIS
2. * Ready to complete released command.	→	DI4: Set_service

DI1: Check_FIS	Check_FIS type and C bit.	
1. Register type, C bit cleared, and SRST set.	→	DSR0: Software_reset_ asserted
2. Register type, C bit cleared, and SRST cleared	→	DI0: Device_idle
3. Register type and C bit set.	→	DI2: Check_command
4. First Party Setup FIS Received	→	DI0 : Device_idle
5. Unexpected FIS type.	→	DI0: Device_idle

DI2: Check_command[1]	Check the command to determine required command protocol.	
1. Non-data command protocol.	→	DND0: Non-data
2. PIO data-in command protocol.	→	DPIOI0: PIO_in
3. PIO data-out command protocol.	→	DPIOO0: PIO_out
4. READ DMA command protocol. (legacy)	→	DDMAI0: DMA_in
5. WRITE DMA command protocol. (legacy)	→	DDMAO0: DMA_out
6. PACKET command protocol.	→	DP0: PACKET
7. READ DMA QUEUED command protocol. (legacy)	→	DDMAQI0: DMA_queued_in
8. WRITE DMA QUEUED command protocol. (legacy)	→	DDMAQIO: DMA_queued_out
9. EXECUTE DEVICE DIAGNOSTIC command protocol.	→	DEDD0: Execute_device_diag
10. DEVICE RESET command protocol.	→	DDR0: Device_reset
11. Command not implemented.	→	DI3: No_command
12. * SERVICE command protocol	→	DI5: Service
NOTE: 1. This state shows transitions for all commands. If a device does not implement any particular command, then that transition should not be processed.		

DI3: No_command	Request transmission of Register FIS with ABRT error.	
1. FIS transmission complete.	→	DI0: Device_idle

* DI4: Set_service	Request transmission of Set Device Bits FIS with SERV set.	
1. FIS transmission complete.	→	DI0: Device_idle

* DI5: Service_test	Test command to see if Register FIS is needed to set DRQ and set Tag.	
1. PACKET PIO data-in or PACKET PIO data-out.	→	DI7: Service_decode
2. Other	→	DI6: Service_send_tag

* DI6: Service_send_tag	Request transmission of Register FIS with BSY=0, DRQ=1, and SC=Tag	
1. FIS transmission complete	→	DI7: Service_decode

* DI7: Service_decode	Check command type to be serviced.	
1. PACKET PIO data-in.	→	DP4: PACKET_PIO_in
2. PACKET PIO data-out.	→	DP6: PACKET_PIO_out
3. PACKET DMA data-in.	→	DP9: PACKET_DMA_in
4. PACKET DMA data-out.	→	DP11: PACKET_DMA_out
5. READ DMA QUEUED.	→	DDMAQI1: Send_data
6. WRITE DMA QUEUED.	→	DDMAQO1: Send_DMA_activate

DI0: Device_Idle: This state is entered when the device has completed the execution of a command protocol, a COMRESET protocol, a software reset protocol, or a queued command has been released.

When in this state, the device is awaiting a command. If queuing is supported, the device may be waiting to acquire data or establish buffer space to complete a queued command

Transition DI0:1: When the device receives an FIS from the Transport layer, the device shall transition to the DI1: Check_FIS state.

*** Transition DI0:2:** When the device is ready to complete the data transfer for a queued command, the device shall transition to the DI4: Set_service state.

DI1: Check_FIS state: This state is entered when the device receives an FIS from the Transport layer.

When in this state, the device shall check the FIS type.

Transition DI1:1: If the FIS type is a Register FIS, the C bit in the FIS is cleared, and the SRST bit in the FIS is set, the device shall transition to the DSR0: Software_reset_asserted state.

Transition DI1:2: If the FIS type is a Register FIS, the C bit in the FIS is cleared, and the SRST bit in the FIS is cleared, the device shall transition to the DI0: Device_idle state.

Transition DI1:3: If the FIS type is a Register FIS and the C bit in the FIS is set, the device shall transition to the DI2: Check_command state.

Transition DI1:4: If the FIS type is a First Party DMA Setup FIS, the device shall inform the Transport layer of the reception of the First Party DMA Setup FIS, and transition to the DI0: Device_idle state.

Transition DI1:5: For any other FIS, the device shall transition to the DI0: Device_idle state.

DI2: Check_command state: This state is entered when the device recognizes that the received Register FIS contains a new command. NOTE: This state shows transitions for all commands. If a device does not implement any particular command, then transition DI2:11 to state DI3:No_command shall be made.

When in this state, the device shall check the command protocol required by the received command.

Transition DI2:1: When the received command is a non-data transfer command, the device shall transition to the DND0: Non-data state.

Transition DI2:2: When the received command is a PIO data-in command, the device shall transition to the DPIOI0: PIO_in state.

Transition DI2:3: When the received command is a PIO data-out command, the device shall transition to the DPIOO0: PIO_out state.

Transition DI2:4: When the received command is a READ DMA command, the device shall transition to the DDMAI0: DMA_in state.

Transition DI2:5: When the received command is a WRITE DMA command, the device shall transition to the DDMAO0: DMA_out state.

Transition DI2:6: When the received command is a PACKET command, the device shall transition to the DP0: PACKET state.

Transition DI2:7: When the received command is a READ DMA QUEUED command, the device shall transition to the DDMAQI0: DMA_queued_in state.

Transition DI2:8: When the received command is a WRITE DMA QUEUED command, the device shall transition to the DDMAQO0: DMA_queued_out state.

Transition DI2:9: When the received command is an EXECUTE DEVICE DIAGNOSTICS command, the device shall transition to the DEDD0: Execute_device_diag state.

Transition DI2:10: When the received command is an RESET DEVICE command, the device shall transition to the DDR0: Device_reset state.

Transition DI2:11: When the received command is not implemented by the device, the device shall transition to the DI3: No_command state.

Transition DI2:12: When the received command is a SERVICE command, the device shall transition to the DI5: Service state.

DI3: No_command state: This state is entered when the device recognizes that the received command is not implemented by the device.

When in this state, the device shall request that the Transport layer transmit a Register FIS with register content as described in the command description in ATA/ATAP-5 (T13/d1321R2) clause 8 and the I bit set to one.

Transition DI2:1: When the Transport layer has transmitted the Register FIS, the device shall transition to the DI0: Device_idle state.

*** DI4: Set_service state:** This state is entered when the ready to complete the data transfer for a queued command.

When in this state, the device shall request that the Transport layer transmit a Set Device Bits FIS with the SERV bit set in the Status register and with all other bits in the Error and Status fields the same as the current contents of the respective registers, and the I bit set to one.

Transition DI4:1: When the Transport layer has transmitted the Set Device Bits FIS, the device shall transition to the DI0: Device_idle state.

*** DI5: Service_test state:** This state is entered when the SERVICE command has been received.

When in this state, the device shall determine the type of command that the device has requested service to complete. The PACKET_PIO command will provide its own register update to set DRQ and send the command tag, but other queued commands require a Register FIS.

Transition DI5:1: When the command to be serviced is a PIO data-in or PIO data-out command, the device shall transition to the DI7: Service_decode state.

Transition DI5:2: When the command to be serviced is neither a PIO data-in nor a PIO data-out command, the device shall transition to the DI6: Service_send_tag state.

*** DI6: Service_send_tag state:** This state is entered when the SERVICE command has been received and sending a Register Device to Host FIS is necessary for the command being serviced.

When in this state, the device shall request that the Transport layer transmit a Register FIS with register contents, including the desired command tag, as described in the command description of the ATA/ATAPI standard for the command being serviced.

Transition DI6:1: When the Transport layer has transmitted the Register FIS, the device shall transition to theDI7: Service_decode state.

*** DI7: Service_decode state:** This state is entered when a register FIS has been transmitted, if necessary to send the register contents, including the desired command tag, in response to a SERVICE command.

When in this state, the device shall again determine the type of command that the device has requested service to complete, and branch to that command's data transfer and completion.

Transition DI7:1: When the command to be serviced is a PIO data-in command, the device shall transition to the DP4: PACKET_PIO_in state.

Transition DI7:2: When the command to be serviced is a PIO data-out command, the device shall transition to the DP6: PACKET_PIO_out state.

Transition DI7:3: When the command to be serviced is a DMA data-in command, the device shall transition to the DP9: PACKET_DMA_in state.

Transition DI7:4: When the command to be serviced is a DMA data-out command, the device shall transition to the DP11: PACKET_DMA_out state.

Transition DI7:5: When the command to be serviced is a READ DMA QUEUED command, the device shall transition to the DDMAQI1: Send_data state.

Transition DI7:6: When the command to be serviced is a WRITE DMA QUEUED command, the device shall transition to the DDMAQO1: Send_DMA_activate state.

9.3 Software reset protocol

When the host sends a Register FIS with a one in the SRST bit position of the Device Control register byte, regardless of the power management mode, the device shall execute the software reset protocol.

DSR0: Software_reset_asserted	Begin initialization and diagnostic execution.	
1. Software reset negated.	→	DSR1: Execute_diagnostics
2. Software reset asserted.	→	DSR0: Software_reset_ asserted

DSR1: Execute_diagnostics	Complete Initialization and diagnostics execution.	
1. Initialization and diagnostics completed successfully.	→	DSR2: Send_good_status
2. Initialization completed and diagnostics failed.	→	DSR3: Send_bad_status

DSR2: Send_good_status	Request transmission of Register FIS with good status.	
1. Register FIS transmitted.	→	DI0: Device_idle

DSR3: Send_bad_status	Request transmission of Register FIS with bad status.	
1. Register FIS transmitted.	→	DI0: Device_idle

DSR0: Software_reset_asserted: This state is entered when a Register FIS is received with the C bit in the FIS cleared to zero and the SRST bit set to one in the Device Control register.

When in this state, the device begins its initialization and diagnostics execution and awaits the clearing of the SRST bit.

Transition DSR0:1: When a Register FIS is received with the C bit in the FIS cleared to zero and the SRST bit cleared to zero in the Device Control register, the device shall transition to the DSR1: Execute_diagnostics state.

Transition DSR0:2: If a Register FIS is received with the C bit in the FIS set to one, or the SRST bit set to one in the Device Control register, the device shall transition to the DSR0: Software_reset_asserted state.

DSR1: Execute_diagnostics: This state is entered when a Register FIS is received with the C bit in the FIS cleared to zero and the SRST bit cleared to zero in the Device Control register.

When in this state, the device completes initialization and execution of its diagnostics.

Transition DSR1:1: When the device has been initialized and the diagnostics successfully completed, the device shall transition to the DSR2: Send_good_status state.

Transition DSR1:2: When the device has been initialized and the diagnostics failed, the device shall transition to the DSR3: Send_bad_status state.

DSR2: Send_good_status: This state is entered when the device has been initialized and the diagnostics successfully completed.

When in this state, the device requests that the Transport layer transmit a Register FIS to the host. If the device does not implement the PACKET command feature set the register content shall be:

Sector Count	01h
Sector Number	01h
Cylinder Low	00h
Cylinder High	00h
Device/Head	00h
Error	01h
Status	00h-70h (see note)
NOTE – Setting of bit 6 thru 4 in the Status register are device specific.	

If the device implements the PACKET command feature set, the register content shall be:

Sector Count	01h
Sector Number	01h
Cylinder Low	14h
Cylinder High	EBh
Device/Head	00h
Error	01h
Status	00h

Transition DSR2:1: When the Transport layer indicates that the Register FIS has been transmitted, the device shall transition to the DI0: Device_Idle state.

DSR3: Send_bad_status: This state is entered when the device has been initialized and the diagnostics failed.

When in this state, the device requests that the Transport layer transmit a Register FIS to the host. If the device does not implement the PACKET command feature set the register content shall be:

Sector Count	01h
Sector Number	01h
Cylinder Low	00h
Cylinder High	00h
Device/Head	00h
Error	00h, 02h-7Fh
Status	00h-70h (see note)
NOTE – Setting of bit 6 thru 4 in the Status register are device specific.	

If the device implements the PACKET command feature set, the register content shall be:

Sector Count	01h
Sector Number	01h
Cylinder Low	14h
Cylinder High	EBh
Device/Head	00h
Error	00h, 02h-7Fh
Status	00h

Transition DSR3:1: When the Transport layer indicates that the Register FIS has been transmitted, the device shall transition to the DI0: Device_Idle state.

9.4 EXECUTE DEVICE DIAGNOSTIC command protocol

This class includes:

- EXECUTE DEVICE DIAGNOSTIC

If the host sends COMRESET before the device has completed executing the EXECUTE DEVICE DIAGNOSTIC protocol, then the device shall immediately start executing the COMRESET protocol form the beginning. If the host asserts SRST in the Device Control register before the device has completed executing the EXECUTE DEVICE DIAGNOSTIC protocol, then the device shall immediately start executing its software reset protocol from the beginning.

DEDD0: Execute_device_diag	Execute diagnostics.	
1. Diagnostics completed successfully.	→	DEDD1: Send_good_status
2. Diagnostics failed.	→	DEDD2: Send_bad_status

DEDD1: Send_good_status	Request transmission of Register FIS with good status.	
1. Register FIS transmitted.	→	DI0: Device_idle

DEDD2: Send_bad_status	Request transmission of Register FIS with bad status.	
1. Register FIS transmitted.	→	DI0: Device_idle

DEDD0: Execute_device_diag: This state is entered when an EXECUTE DEVICE DIAGNOSTIC command is received.

When in this state, the device executes its diagnostics.

Transition DEDD0:1: When the device successfully completed the diagnostics, the device shall transition to the DEDD1: Send_good_status state.

Transition DEDD1:2: When the device has failed the diagnostics, the device shall transition to the DEDD2: Send_bad_status state.

DEDD1: Send_good_status: This state is entered when the device has successfully completed the diagnostics.

When in this state, the device shall request that the Transport layer transmit a Register FIS to the host, with the I bit set to one. If the device does not implement the PACKET command feature set the register content shall be:

Sector Count	01h
Sector Number	01h
Cylinder Low	00h
Cylinder High	00h
Device/Head	00h
Error	01h
Status	00h-70h (see note)
NOTE – Setting of bit 6 thru 4 in the Status register are device specific.	

If the device implements the PACKET command feature set, the register content shall be:

Sector Count	01h
Sector Number	01h
Cylinder Low	14h
Cylinder High	EBh
Device/Head	00h
Error	01h
Status	00h

Transition DEDD1:1: When the Transport layer indicates that the Register FIS has been transmitted, the device shall transition to the DI0: Device_Idle state.

DEDD2: Send_bad_status: This state is entered when the device has been initialized and the diagnostics failed.

When in this state, the device shall request that the Transport layer transmit a Register FIS to the host, with the I bit set to one. If the device does not implement the PACKET command feature set the register content shall be:

Sector Count	01h
Sector Number	01h
Cylinder Low	00h
Cylinder High	00h
Device/Head	00h
Error	00h, 02h-7Fh
Status	00h-70h (see note)
NOTE – Setting of bit 6 thru 4 in the Status register are device specific.	

If the device implements the PACKET command feature set, the register content shall be:

Sector Count	01h
Sector Number	01h
Cylinder Low	14h
Cylinder High	EBh
Device/Head	00h
Error	00h, 02h-7Fh
Status	00h

Transition DEDD2:1: When the Transport layer indicates that the Register FIS has been transmitted, the device shall transition to the DI0: Device_Idle state.

9.5 DEVICE RESET command protocol

This class includes:

- DEVICE RESET

If the host sends COMRESET before the device has completed executing the DEVICE RESET protocol, then the device shall immediately start executing the COMRESET protocol from the beginning. If the host asserts SRST in the Device Control register before the device has completed executing the DEVICE RESET protocol, then the device shall immediately start executing its software reset protocol from the beginning.

DDR0: Device_reset	Stop execution and background activity.	
1. Activity ceased.	→	DDR1: Send_good_status

DDR1: Send_good_status	Request transmission of Register FIS with good status.	
1. Register FIS transmitted.	→	DI0: Device_idle

DDR0: Device_reset: This state is entered when an DEVICE RESET command is received.

When in this state, the device stops any execution or activity in progress.

Transition DDR0:1: When the device has ceased any execution or activity and has completed its internal diagnostics, the device shall transition to the DDR1: Send_good_status state.

DDR1: Send_good_status: This state is entered when the device has been initialized and the diagnostics successfully completed.

When in this state, the device requests that the Transport layer transmit a Register FIS to the host.

The register content shall be:

Sector Count	01h
Sector Number	01h
Cylinder Low	14h
Cylinder High	EBh
Device/Head	00h
Error	01h
Status	00h

Transition DDR1:1: When the Transport layer indicates that the Register FIS has been transmitted, the device shall transition to the DI0: Device_Idle state.

9.6 Non-data command protocol

This class includes:

- CFA ERASE SECTORS
- CFA REQUEST EXTENDED ERROR CODE
- CHECK POWER MODE
- FLUSH CACHE
- FLUSH CACHE EXT
- GET MEDIA STATUS
- IDLE
- IDLE IMMEDIATE
- INITIALIZE DEVICE PARAMETERS
- MEDIA EJECT
- MEDIA LOCK
- MEDIA UNLOCK

- NOP
- READ NATIVE MAX ADDRESS
- READ NATIVE MAX ADDRESS EXT
- READ VERIFY SECTOR(S)
- SECURITY ERASE PREPARE
- SECURITY FREEZE LOCK
- SEEK
- SET FEATURES
- SET MAX ADDRESS
- SET MAX ADDRESS EXT
- SET MULTIPLE MODE
- SLEEP
- SMART DISABLE OPERATION
- SMART ENABLE/DISABLE AUTOSAVE
- SMART ENABLE OPERATION
- SMART EXECUTE OFFLINE IMMEDIATE
- SMART RETURN STATUS
- STANDBY
- STANDBY IMMEDIATE

Execution of these commands involves no data transfer. See the NOP command description and the SLEEP command description for additional protocol requirements.

DND0: Non-data	Execute Non-data command.		
	1. Command execution complete.	→	DND1: Send_status

DND1: Send_status	Request transmission of a Register FIS.		
	1. FIS transmission complete.	→	DI0: Device_idle

DND0: Non-data State: This state is entered when a received command is a non-data command.

When in this state, the device shall execute the requested command if supported.

Transition DND0: 1: When command execution completes, the device shall transition to the DND1: Send_status state.

DND1: Send_status State: This state is entered when the execution of the non-data command has been completed.

When in this state, the device shall request that the Transport layer transmit a Register FIS with register content as described in the command description in ATA/ATAP-5 (T13/d1321R2) clause 8 and the I bit set to one.

Transition DND1:1: When the FIS has been transmitted, then the device shall transition to the DI0: Device_idle state.

9.7 PIO data-in command protocol

This class includes:

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

Execution of this class of command includes the PIO transfer of one or more blocks of data from the device to the host.

DPIOI0: PIO_in	Prepare a DRQ data block for transfer to the host.		
	1. DRQ data block ready to transfer.	→	DPIOI1: Send_PIO_setup
	2. Command aborted due to error.	→	DPIOI3: Error_status

DPIOI1: Send_PIO_setup	Request transmission of a PIO Setup FIS to host.		
	1. PIO Setup FIS transmitted.	→	DPIOI2: Transmit_data

DPIOI2: Transmit_data	Request transmission of a Data FIS to host.		
	1. Data FIS transmitted, no more data transfer required for this command.	→	DI0: Device_idle
	2. Data FIS transmitted, more data transfer required for this command, or 8KB transmitted.	→	DPIOI0: PIO_in

DPIOI3: Error_status	Request transmission of a Register - Device to Host FIS.		
	1. FIS transmission complete.	→	DI0: Device_idle

DPIOI0: PIO_in State: This state is entered when the device receives a PIO data-in command or the transmission of one or more additional DRQ data blocks is required to complete the command.

When in this state, device shall prepare a DRQ data block for transfer to the host.

Transition DPIOI0:1: When the device has a DRQ data block ready to transfer, the device shall transition to the DPIOI1: Send_PIO_setup state.

Transition DPIOI0:2: When the device has encountered an error that causes the command to abort before completing the transfer of the requested data, the device shall transition to the DPIOI3: Error_status state.

DPIOI1: Send_PIO_setup: This state is entered when the device is ready to transmit a DRQ data block from the host.

When in this state, the device shall request that the Transport layer transmit a PIO Setup FIS. The initial status shall have BSY cleared to zero and DRQ set to one and with register content as described in the command description in ATA/ATAP-5 (T13/d1321R2) clause 8. The I bit shall be set. If this is the last DRQ data block requested by the command, the ending status shall have BSY cleared to zero and DRQ cleared to zero. If this is not the last data block requested by the command, the ending status shall have BSY set to one and DRQ cleared to zero.

Transition DPIOI1:1: When the PIO Setup FIS has been transferred, the device shall transition to the DPIOI2: Transmit_data state.

DPIOI2: Transmit_data: This state is entered when the device has transmitted a PIO Setup FIS to the host.

When in this state, the device shall request that the Transport layer transmit a Data FIS containing the DRQ data block.

Transition DPIOI2:1: When the Data FIS has been transferred and all data requested by this command have been transferred, the device shall transition to the DI0: Device_idle.

Transition DPIOI2:2: When the Data FIS has been transferred but all data requested by this command has not been transferred, or the 8KB transfer limit has been reached, then the device shall transition to the DPIOI0: PIO_in state.

DPIOI3: Error_status: This state is entered when the device has encountered an error that causes the command to abort before completing the transfer of the requested data.

When in this state, the device shall request that the Transport layer transmit a Register FIS with register content as described in the command description in ATA/ATAP-5 (T13/d1321R2) clause 8 and the I bit set to one.

Transition DPIOI3:1: When the FIS has been transmitted, the device shall transition to the DI0: Device_idle state.

9.8 PIO data-out command protocol

This class includes:

- CFA WRITE MULTIPLE WITHOUT ERASE
- CFA WRITE SECTORS WITHOUT ERASE
- DOWNLOAD MICROCODE
- SECURITY DISABLE PASSWORD
- SECURITY ERASE UNIT
- SECURITY SET PASSWORD
- SECURITY UNLOCK
- SMART WRITE LOG SECTOR
- WRITE BUFFER
- WRITE MULTIPLE
- WRITE MULTIPLE EXT
- WRITE SECTOR(S)
- WRITE SECTOR(S) EXT

Execution of this class of command includes the PIO transfer of one or more blocks of data from the host to the device.

DPIOO0: PIO_out	Prepare to receive DRQ data block transfer from the host.		
	1. Ready to receive DRQ data block transfer.	→	DPIOO1: Send_PIO_setup
	2. All DRQ data blocks received or command aborted due to error.	→	DPIOO3: Send_status

DPIOO1: Send_PIO_setup	Request transmission of a PIO Setup FIS to host.		
	1. PIO Setup FIS transmitted.	→	DPIOO2: Receive_data

DPIOO2: Receive_data	Receive Data FIS from the Transport layer.		
	1. Data FIS received.	→	DPIOO0: PIO_out

DPIOO3: Send_status	Request transmission of a Register - Device to Host FIS.		
	1. FIS transmission complete.	→	DI0: Device_idle

DPIOI0: PIO_out State: This state is entered when the device receives a PIO data-out command or the receipt of one or more DRQ data blocks is required to complete this command.

When in this state, device shall prepare to receive a DRQ data block transfer from the host.

Transition DPIOO0:1: When the device is ready to receive a DRQ data block, the device shall transition to the DPIOO1: Send_PIO_setup state.

Transition DPIOO0:2: When the device has received all DRQ data blocks requested by this command or the device has encountered an error that causes the command to abort before completing the transfer of the requested data, then the device shall transition to the DPIOO3: Send_status state.

DPIOO1: Send_PIO_setup: This state is entered when the device is ready to receive a DRQ data block from the host.

When in this state, the device shall request that the Transport layer transmit a PIO Setup FIS. The initial status shall have BSY cleared to zero and DRQ set to one. If this is the first DRQ data block for this command, the I bit shall be cleared to zero. If this is not the first DRQ data block for this command, the I bit shall be set to one. The ending status shall have BSY set to one and DRQ cleared to zero. The byte count for the DRQ data block shall be indicated.

Transition DPIOO1:1: When the PIO Setup FIS has been transferred, the device shall transition to the DPIOO2: Receive_data state.

DPIOO2:Receive_data: This state is entered when the device has transmitted a PIO Setup FIS to the host.

When in this state, the device shall receive the requested Data FIS from the Transport layer.

Transition DPIOO2:1: When the Data FIS has been received, the device shall transition to the DPIOO0: PIO_out state.

DPIOO3: Send_status: This state is entered when the device has received all DRQ data blocks requested by this command or the device has encountered an error that causes the command to abort before completing the transfer of the requested data.

When in this state, the device shall request that the Transport layer transmit a Register FIS with register content as described in the command description in ATA/ATAP-5 (T13/d1321R2) clause 8 and the I bit set to one.

Transition DPIOO3:1: When the FIS has been transmitted, the device shall transition to the DI0: Device_idle state.

9.9 DMA data in command protocol

This class includes:

- READ DMA
- READ DMA EXT

Execution of this class of command includes the transfer of one or more blocks of data from the device to the host using DMA transfer.

DDMAI0: DMA_in	Prepare data for the transfer of a Data FIS.	
1. Data for Data FIS ready to transfer.	→	DDMAI1: Send_data
2. Command completed or aborted due to error.	→	DDMAI2: Send_status

DDMAI1: Send_data	Request transmission of a Data FIS to host.	
1. Data FIS transmitted. No more data transfer required for this command, or 8KB transmitted.	→	DDMAI0: DMA_in

DDMAI2: Send_status	Request transmission of a Register FIS to host.	
1. Register FIS transmitted.	→	DI0: Device_idle

DDMAI0: DMA_in State: This state is entered when the device receives a DMA data-in command or the transmission of one or more data FIS is required to complete the command.

When in this state, device shall prepare the data for transfer of a data FIS to the host.

Transition DDMAI0:1: When the device has the data ready to transfer a data FIS, the device shall transition to the DDMAI1: Send_data state.

Transition DDMAI0:2: When the device has transferred all of the data requested by this command or has encountered an error that causes the command to abort before completing the transfer of the requested data, then the device shall transition to the DDMAI2: Send_status state.

DDMAI1: Send_data: This state is entered when the device has the data ready to transfer a data FIS to the host.

When in this state, the device shall request that the Transport layer transmit a data FIS containing the data. The device command layer shall request a data FIS size of no more than 8KB.

Transition DDMAI1:1: When the data FIS has been transferred, the device shall transition to the DMAOI0: DMA_in state.

DDMAI2: Send_status: This state is entered when the device has transferred all of the data requested by the command or has encountered an error that causes the command to abort before completing the transfer of the requested data.

When in this state, the device shall request that the Transport layer transmit a Register FIS with register content as described in the command description in ATA/ATAP-5 (T13/d1321R2) clause 8 and the I bit set to one.

Transition DDMAI2:1: When the FIS has been transmitted, the device shall transition to the DI0: Device_idle state.

9.10 DMA data out command protocol

This class includes:

- WRITE DMA
- WRITE DMA EXT

Execution of this class of command includes the transfer of one or more blocks of data from the host to the device using DMA transfer. A single interrupt is issued at the completion of the successful transfer of all data required by the command.

DDMAO0: DMA_out	Prepare to receive a Data FIS from the host.	
1. Ready to receive Data FIS.	→	DDMAO1: Send_DMA_activate
2. All data requested for this command received or command aborted due to error.	→	DDMAO3: Send_status

DDMAO1: Send_DMA_activate	Request transmission of a DMA Activate FIS to host.	
1. DMA Activate FIS transmitted.	→	DDMAO2: Receive_data

DDMAO2: Receive_data	Receive Data FIS from the Transport layer.	
1. Data FIS received.	→	DDMAO0: DMA_out

DDMAO3: Send_status	Request transmission of a Register FIS to host.	
1. Register FIS transmitted.	→	DI0: Device_idle

DDMAO0: DMA_out State: This state is entered when the device receives a DMA data-out command or the receipt of one or more Data FIS is required to complete this command.

When in this state, device shall prepare to receive a Data FIS from the host.

Transition DDMAO0:1: When the device is ready to receive a Data FIS, the device shall transition to the DDMAO1: Send_DMA_activate state.

Transition DDMAO0:2: When the device has received all the data requested by this command or the device has encountered an error that causes the command to abort before completing the transfer of the requested data, then the device shall transition to the DDMAO3: Send_status state.

DDMAO1: Send_DMA_activate: This state is entered when the device is ready to receive a Data FIS from the host.

When in this state, the device shall request that the Transport layer transmit a DMA Activate FIS.

Transition DDMAO1:1: When the DMA Activate FIS has been transferred, the device shall transition to the DDMAO2: Receive_data state.

DDMAO2: Receive_data: This state is entered when the device transmitted a DMA Activate FIS to the host.

When in this state, the device shall receive the requested Data FIS from the Transport layer.

Transition DDMAO2:1: When the Data FIS has been received, the device shall transition to the DDMAO0: DMA_out state.

DDMAO3: Send_status: This state is entered when the device has received all the data requested by this command or the device has encountered an error that causes the command to abort before completing the transfer of the requested data.

When in this state, the device shall request that the Transport layer transmit a Register FIS with register content as described in the command description in ATA/ATAP-5 (T13/d1321R2) clause 8 and the I bit set to one.

Transition DDMAO3:1: When the FIS has been transmitted, the device shall transition to the DI0: Device_idle state.

9.11 PACKET protocol

This class includes:

- PACKET

States marked with an * are only utilized when queuing is implemented

DP0: PACKET	Request transmission of a PIO Setup FIS.		
	1. FIS transmission complete.	→	DP1: Receive_command

DP1: Receive_command	Receive Data FIS containing command packet.		
	1. FIS receiption complete.	→	DP2: Check_command

DP2: Check_command	Determine the protocol required for the received command.		
	1. Non-data command.	→	DP3: PACKET_non-data
	2. PIO data-in command.	→	DP4: PACKET_PIO_in
	3. PIO data-out command.	→	DP6: PACKET_PIO_out
	4. DMA data-in command	→	DP9: PACKET_DMA_in
	5. DMA data-out command	→	DP11: PACKET_DMA_out

DP3: PACKET_non-data	Execute Non-data command.		
	1. Command execution complete.	→	DP14: Send_status

DP4: PACKET_PIO_in	Prepare a DRQ data block for transfer to the host.		
	1. DRQ data block ready to transfer.	→	DP4a: PIO_in_setup
	2. Transfer complete or command aborted due to error.	→	DP14: Send_status
	3. * DRQ block is not ready for immediate transfer	→	DP15: Release

DP4a: PIO_in_setup	Request transmission of a PIO Setup FIS to host.		
	1. PIO Setup FIS transmitted.	→	DP5: Send_PIO_data

DP5: Send_PIO_data	Request transmission of a Data FIS to host.		
	1. Data FIS transmitted.	→	DP4: PACKET_PIO_in

DP6: PACKET_PIO_out	Prepare to receive DRQ data block from the host.		
	1. Ready to receive DRQ data block transfer.	→	DP7: PIO_out_setup
	2. All DRQ data blocks received or command aborted due to error.	→	DP14: Send_status
	3. * Not ready to accept DRQ block immediately.	→	DP15: Release

DP7: PIO_out_setup	Request transmission of a PIO Setup FIS to host.	
1. PIO Setup FIS transmitted.	→	DP8: Receive_PIO_data

DP8: Receive_PIO_data	Receive Data FIS from the Transport layer.	
1. Data FIS received.	→	DP6: PACKET_PIO_out

DP9: PACKET_DMA_in	Prepare data for the transfer of a Data FIS.	
1. Data for Data FIS ready to transfer.	→	DP10: Send_DMA_data
2. Command completed or aborted due to error.	→	DP14: Send_status
3. * Data is not ready for immediate transfer.	→	DP15: Release

DP10: Send_DMA_data	Request transmission of a Data FIS to host.	
1. Data FIS transmitted. No more data transfer required for this command, or 8KB transmitted.	→	DP9: PACKET_DMA_IN

DP11: PACKET_DMA_out	Prepare to receive a Data FIS from the host.	
1. Ready to receive Data FIS.	→	DP12: Send_DMA_activate
2. All data requested for this command received or command aborted due to error.	→	DP14: Send_status
3. * Not ready for immediate transfer.	→	DP15: Release

DP12: Send_DMA_activate	Request transmission of a DMA Activate FIS to host.	
1. DMA Activate FIS transmitted.	→	DP13: Receive_DMA_data

DP13: Receive_DMA_data	Receive Data FIS from the Transport layer.	
1. Data FIS received.	→	DP11: PACKET_DMA_out

DP14: Send_status	Request transmission of a Register FIS.	
1. FIS transmission complete.	→	DI0: Device_idle

* DP15: Release	Request transmission of a Register FIS.	
1. FIS transmission complete.	→	DI0: Device_idle

DP0: PACKET: This state is entered when the device receives a PACKET command.

When in this state, the device shall request that the Transport layer transmit a PIO Setup FIS to acquire the command packet associated with this command. The initial status shall have BSY cleared to zero and DRQ set to one. The I bit shall be set to zero. The ending status shall have BSY set to one and DRQ cleared to zero. The byte count for the DRQ data block shall be indicated.

Transition DP0:1: When the PIO Setup FIS has been transferred, the device shall transition to the DP1: Receive_command state.

DP1: Receive_command: This state is entered when the device transmitted a PIO Setup FIS to the host to get the command packet.

When in this state, the device shall receive the requested Data FIS from the Transport layer.

Transition DP1:1: When the Data FIS has been received, the device shall transition to the DP2: Check_command state.

DP2: Check_command: This state is entered when the Data FIS containing the command packet has been received.

When in this state, the device shall determine the protocol for the command contained in the command packet.

Transition DP2:1: When the command is a non-data transfer command, the device shall transition to the DP3: PACKET_non-data state.

Transition DP2:2: When the command is a PIO data-in transfer command, the device shall transition to the DP4: PACKET_PIO_in state.

Transition DP2:3: When the command is a PIO data-out transfer command, the device shall transition to the DP6: PACKET_PIO_out state.

Transition DP2:4: When the command is a DMA data-in transfer command, the device shall transition to the DP9: PACKET_DMA_in state.

Transition DP2:5: When the command is a DMA data-out transfer command, the device shall transition to the DP11: PACKET_DMA_out state.

DP3: PACKET_non-data State: This state is entered when a received command is a non-data command.

When in this state, the device shall execute the requested command.

Transition DP3:1: When command execution completes, the device shall transition to the DP14: Send_status state.

DP4: PACKET_PIO_in State: This state is entered when the device receives a PIO data-in command or the transmission of one or more DRQ data blocks is required to complete the command.

When in this state, device shall prepare a DRQ data block for transfer to the host.

Transition DP4:1: When the device has a DRQ data block ready to transfer, the device shall transition to the DP4a: PIO_in_setup.

Transition DP4:2: When all of the data requested by this command has been transferred or the device has encountered an error that causes the command to abort before completing the transfer of the requested data, then the device shall transition to the DP14: Send_status state.

*** Transition DP4:3:** When the device supports overlap and queuing and does not have a DRQ data block ready to transfer immediately, the device shall transition to the DP15: Release state.

DP4a: PIO_in_setup: This state is entered when the device is ready to transfer a DRQ block to the host.

When in this state, the device shall request that the Transport layer transmit a PIO Setup FIS. The initial status shall have BSY cleared to zero and DRQ set to one. The I bit shall be set to one. The ending status shall have BSY set to one and DRQ cleared to zero. The byte count for the DRQ data block shall be indicated.

Transition DP4a:1: When the PIO Setup FIS has been transferred, the device shall transition to the DP5:SendPIO_data state.

DP5:Send_PIO_data: This state is entered when the device is ready to transfer a DRQ data block to the host.

When in this state, the device shall request that the Transport layer transmit a Data FIS containing the DRQ data block.

Transition DP5:1: When the Data FIS has been transferred, the device shall transition to the DP4: PACKET_PIO_in state.

DP6: PACKET_PIO_out State: This state is entered when the device receives a PIO data-out command or the receipt of one or more DRQ data blocks is required to complete the command.

When in this state, device shall prepare to receive a DRQ data block transfer from the host.

Transition DP6:1: When the device is ready to receive a DRQ data block transfer, the device shall transition to the DP7: PIO_out_setup state.

Transition DP6:2: When the device has received all DRQ data blocks requested by this command or the device has encountered an error that causes the command to abort before completing the transfer of the requested data, then the device shall transition to the DP14: Send_status state.

*** Transition DP6:3:** When the device supports overlap and queuing and can not accept a DRQ data block immediately, the device shall transition to the DP15: Release state.

DP7: PIO_out_setup: This state is entered when the device is ready to receive a DRQ data block from the host.

When in this state, the device shall request that the Transport layer transmit a PIO Setup FIS. The initial status shall have BSY cleared to zero and DRQ set to one. The I bit shall be set to one. The ending status shall have BSY set to one and DRQ cleared to zero. The byte count for the DRQ data block shall be indicated.

Transition DP7:1: When the PIO Setup FIS has been transferred, the device shall transition to the DP8: Receive_PIO_data state.

DP8: Receive_PIO_data: This state is entered when the device transmitted a PIO Setup FIS to the host.

When in this state, the device shall receive the requested Data FIS from the Transport layer.

Transition DP8:1: When the Data FIS has been received, the device shall transition to the DP6: PACKET_PIO_out state.

DP9: PACKET_DMA_in State: This state is entered when the device receives a DMA data-in command or the transmission of one or more Data FIS is required to complete the command.

When in this state, device shall prepare the data for transfer of a Data FIS to the host.

Transition DP9:1: When the device has the data ready to transfer a Data FIS, the device shall transition to the DP10: Send_DMA_data state.

Transition DP9:2: When the device has transferred all of the data requested by this command or has encountered an error that causes the command to abort before completing the transfer of the requested data, then the device shall transition to the DP14: Send_status state.

*** Transition DP9:3:** When the device supports overlap and queuing and does not have data ready to transfer immediately, the device shall transition to the DP15: Release state.

DP10: Send_DMA_data: This state is entered when the device has the data ready to transfer a Data FIS to the host.

When in this state, the device shall request that the Transport layer transmit a Data FIS containing the data.

Transition DP10:1: When the Data FIS has been transferred, the device shall transition to the DP9: PACKET_DMA_in state. The device command layer shall request a data FIS size of no more than 8KB.

DP11: PACKET_DMA_out State: This state is entered when the device receives a DMA data-out command or the receipt of one or more Data FIS is required to complete the command.

When in this state, device shall prepare to receive a Data FIS from the host.

Transition DP11:1: When the device is ready to receive a Data FIS, the device shall transition to the DP12: Send_DMA_activate state.

Transition DP11:2: When the device has received all the data requested by this command or the device has encountered an error that causes the command to abort before completing the transfer of the requested data, then the device shall transition to the DP14: Send_status state.

*** Transition DP11:3:** When the device supports overlap and queuing and can not accept a Data FIS immediately, the device shall transition to the DP15: Release state.

DP12: Send_DMA_activate: This state is entered when the device is ready to receive a Data FIS from the host.

When in this state, the device shall request that the Transport layer transmit a DMA Activate FIS.

Transition DP12:1: When the DMA Activate FIS has been transferred, the device shall transition to the DP13: Receive_DMA_data state.

DP13: Receive_DMA_data: This state is entered when the device transmitted a DMA Activate FIS to the host.

When in this state, the device shall receive the requested Data FIS from the Transport layer.

Transition DP13:1: When the Data FIS has been received, the device shall transition to the DP11: PACKET_DMA_out state.

DP14: Send_status: This state is entered when the device has received all the data requested by this command or the device has encountered an error that causes the command to abort before completing the transfer of the requested data.

When in this state, the device shall request that the Transport layer transmit a Register FIS with register content as described in the command description in ATA/ATAP-5 (T13/d1321R2) clause 8 and the I bit set to one.

Transition DP14:1: When the FIS has been transmitted, then the device shall transition to the DI0: Device_idle state.

*** DP15: Release:** This state is entered when the device is not able to do a data transfer immediately.

When in this state, the device shall request that the Transport layer transmit a Register FIS with register content as described in the command description in ATA/ATAPI-5 (T13/d1321R2) clause 8, with the REL bit set to one, and, if the bus release interrupt has been enabled by a previous Set Features Command, with the I bit set to one.

Transition DP15:1: When the FIS has been transmitted, then the device shall transition to the DI0: Device_idle state.

9.12 READ DMA QUEUED command protocol

This class includes:

- READ DMA QUEUED
- READ DMA QUEUED EXT

Execution of this class of command includes the transfer of one or more blocks of data 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 controller prior to transferring data. When data transfer is begun, all data for the request shall be transferred without a bus release.

DDMAQI0: DMA_queued_in	Determine whether to transfer or release.		
	1. Data for Data FIS ready to transfer.	→	DDMAQI1: Send_data
	2. Command aborted due to error.	→	DDMAQI3: Send_status
	3. Bus release	→	DDMAQI4: Release

DDMAQI1: Send_data	Request transmission of a Data FIS to host.		
	1. Data FIS transmitted. No more data transfer required for this command, or 8KB transmitted.	→	DDMAQI2: Prepare_data

DDMAQI2: Prepare_data	Prepare data for the next Data FIS.		
	1. Data ready.	→	DDMAQI1: Send _data
	2. Command complete or aborted due to error.	→	DDMAQI3: Send_status

DDMAQI3: Send_status	Request transmission of a Register FIS to host.		
	1. Register FIS transmitted.	→	DI0: Device_idle

DDMAQI4: Release	Request transmission of a Register FIS to host.	
1. Register FIS transmitted.	→	DI0: Device_idle

DDMAQI0: DMA_queued_in State: This state is entered when the device receives a READ DMA QUEUED command.

When in this state, device shall determine if the requested data is ready to transfer to the host.

Transition DDMAQI0:1: When the device has the requested data ready to transfer a Data FIS immediately, the device shall transition to the DDMAQI1: Send_data state.

Transition DDMAQI0:2: When the device has encountered an error that causes the command to abort before completing the transfer of the requested data, the device shall transition to the DDMAQI3: Send_status state.

Transition DDMAQI0:3: When the device does not have the requested data ready to transfer a Data FIS immediately, the device shall transition to the DDMAQI4: Release state.

DDMAQI1: Send_data: This state is entered when the device has the data ready to transfer a Data FIS to the host.

When in this state, the device shall request that the Transport layer transmit a Data FIS containing the data.
Transition DDMAQI1:1: When the Data FIS has been transferred, the device shall transition to the DDMAQI2: Prepare_data state. The device command layer shall request a data FIS size of no more than 8KB.

DDMAQI2: Prepare_data: This state is entered when the device has completed the transfer a Data FIS to the host.

When in this state, the device shall prepare the data for the next Data FIS.

Transition DDMAQI2:1: When data is ready for the Data FIS, the device shall transition to the DDMAQI1: Send_data state.

Transition DDMAQI2:2: When all data requested for the command has been transmitted or an error has been encountered that causes the command to abort before completing the transfer of the requested data, the device shall transition to the DDMAQI3: Send_status state.

DDMAQI3: Send_status: This state is entered when the device has transferred all of the data requested by the command or has encountered an error that causes the command to abort before completing the transfer of the requested data.

When in this state, the device shall request that the Transport layer transmit a Register FIS with register content as described in the command description in ATA/ATAP-5 (T13/d1321R2) clause 8 and the I bit set to one.

Transition DDMAQI3:1: When the FIS has been transmitted, the device shall transition to the DI0: Device_idle state.

DDMAQI4: Release: This state is entered when the device does not have the requested data available for immediate transfer.

When in this state, the device shall request that the Transport layer transmit a Register FIS with the REL bit set to one, with register content as described in the command description in ATA/ATAPI-5 (T13/d1321R2) clause 8, and, if the bus release interrupt has been enabled by a previous Set Features Command, with the I bit set to one.

Transition DDMAQI4:1: When the FIS has been transmitted, then the device shall transition to the DI0: Device_idle state.

9.13 WRITE DMA QUEUED command protocol

This class includes:

- WRITE DMA QUEUED
- WRITE DMA QUEUED EXT

Execution of this class of command includes the transfer of one or more blocks of data 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 controller prior to transferring data. When data transfer is begun, all data for the request shall be transferred without a bus release.

DDMAQO0: DMA-queued_out	Determine whether to transfer or release.	
1. Ready to accept Data FIS	→	DDMAQO1: Send_DMA_activate
2. Command due to error.	→	DDMAQO4: Send_status
3. Bus release	→	DDMAQO5: Release

DDMAQO1: Send_DMA_activate	Request transmission of a DMA Activate FIS to host.	
1. DMA Activate FIS transmitted.	→	DDMAQO2: Receive_data

DDMAQO2: Receive_data	Receive Data FIS from the Transport layer.	
1. Data FIS received.	→	DDMAQO3: Prepare_data_buffer

DDMAQO3: Prepare_data_buffer	Prepare to receive the next Data FIS.	
1. Ready to receive.	→	DDMAQO1: Send DMA_activate
2. Command complete or aborted due to error.	→	DDMAQO4: Send_status

DDMAQO4: Send_status	Request transmission of a Register FIS to host.	
1. Register FIS transmitted.	→	DI0: Device_idle

DDMAQO5: Release	Request transmission of a Register FIS to host.	
1. Register FIS transmitted.	→	DI0: Device_idle

DDMAQO0: DMA_queued_out State: This state is entered when the device receives a WRITE DMA QUEUED command.

When in this state, device shall determine if it is ready to accept the requested data from the host.

Transition DDMAQO0: 1: When the device is ready to receive a Data FIS immediately, the device shall transition to the DDMAQO1: Send_DMA_activate state.

Transition DDMAQO0:2: When the device has encountered an error that causes the command to abort before completing the transfer of the requested data, the device shall transition to the DDMAQO4: Send_status state.

Transition DDMAQO0:3: When the device does not have the requested data ready to transfer a Data FIS immediately, the device shall transition to the DDMAQO5: Release state.

DDMAQO1:Send_DMA_activate: This state is entered when the device is ready to receive a Data FIS from the host.

When in this state, the device shall request that the Transport layer transmit a DMA Activate FIS.

Transition DDMAQO1:1: When the DMA Activate FIS has been transferred, the device shall transition to the DDMAQO2: Receive_data state.

DDMAQO2:Receive_data: This state is entered when the device transmitted a DMA Activate FIS to the host.

When in this state, the device shall receive the requested Data FIS from the Transport layer.

Transition DDMAQO2:1: When the Data FIS has been received, the device shall transition to the DDMAQO3: Prepare_data_buffer state.

DDMAQO3: Prepare_data_buffer: This state is entered when the device has completed receiving a Data FIS from the host.

When in this state, the device shall prepare for receipt of the next Data FIS.

Transition DDMAQO3:1: When ready to receive the Data FIS, the device shall transition to the DDMAQO1: Send_DMA_activate state.

Transition DDMAQO3:2: When all data requested for the command has been transmitted or an error has been encountered that causes the command to abort before completing the transfer of the requested data, the device shall transition to the DDMAQO4: Send_status state.

DDMAQO4: Send_status: This state is entered when the device has transferred all of the data requested by the command or has encountered an error that causes the command to abort before completing the transfer of the requested data.

When in this state, the device shall request that the Transport layer transmit a Register FIS with register content as described in the command description in ATA/ATAP-5 (T13/d1321R2) clause 8 and the I bit set to one.

Transition DDMAQO4:1: When the FIS has been transmitted, then the device shall transition to the DI0: Device_idle state.

DDMAQO5: Release: This state is entered when the device cannot receive the requested data immediately.

When in this state, the device shall request that the Transport layer transmit a Register FIS with REL set to one, with register content as described in the command description in ATA/ATAPI-5 (T13/d1321R2) clause 8, and, if the bus release interrupt has been enabled by a previous Set Features Command, with the I bit set to one.

Transition DDMAQO5:1: When the FIS has been transmitted, then the device shall transition to the DI0: Device_idle state.

10 Host adapter register interface

Serial ATA host adapters include an additional block of registers mapped separately and independently from the ATA Command Block Registers for reporting additional status and error information and to allow control of capabilities unique to Serial ATA. These additional registers, referred to as the Serial ATA Status and Control Registers (SCR's) are organized as 16 contiguous 32-bit registers. The base address and mapping scheme for these registers is defined by the specific host adapter implementation – for example, PCI controller implementations may map the SCR's using the PCI mapping capabilities. Table 27. – SCR definition illustrates the overall organization of the Serial ATA register interface including both the ATA Command Block Registers and the Status and Control registers. Legacy software does not make use of the Serial ATA Status and Control registers. The Serial ATA Status and Control register are associated with the serial interface and are independent of any master/slave emulation the host adapter may implement.

Table 27. – SCR definition

ATA Command Block and Control Block registers

	A2	A1	A0	Read	Write
CS 0	0	0	0	Data	Data
	0	0	1	Error	Features
	0	1	0	Sector Count [15:8] [7:0]	Sector Count [15:8] [7:0]
	0	1	1	Sector Number [31:24] [7:0]	Sector Number [31:24] [7:0]
	1	0	0	Cylinder Low [39:32] [15:8]	Cylinder Low [39:32] [15:8]
	1	0	1	Cylinder High [47:40] [23:16]	Cylinder High [47:40] [23:16]
	1	1	0	Device/Head	Device/Head
	1	1	1	Status	Command
CS 1	1	1	0	Alternate Status	Device Control

Serial ATA Status and Control registers

SATA register	0	SATA Status/Control
SATA register	1	SATA Status/Control
…	…	SATA Status/Control
SATA register	14	SATA Status/Control
SATA register	15	SATA Status/Control

10.1 SStatus, SError and SControl registers

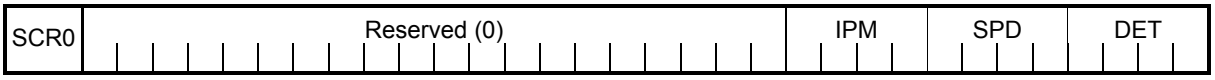
Serial ATA provides an additional block of registers to control the interface and to retrieve interface state information. There are 16 contiguous registers allocated of which the first three are defined and the remaining 13 are reserved for future definition. Table 28. – SCR Definition defines the Serial ATA Status and Control registers.

Table 28. – SCR Definition

SCR[0]	SStatus register
SCR[1]	SError register
SCR[2]	SControl register
SCR[3]	Reserved
…	…
SCR[15]	Reserved

10.1.1 SStatus register

The Serial ATA interface Status register - SStatus - is a 32-bit read-only register that conveys the current state of the interface and host adapter. The register conveys the interface state at the time it is read and is updated continuously and asynchronously by the host adapter. Writes to the register have no effect.



DET The DET value indicates the interface device detection and Phy state.
- 0000 No device detected and Phy commnunication not established
- 0001 Device presence detected but Phy communication not established
- 0011 Device presence detected and Phy communication established
- 0100 Phy in offline mode as a result of the interface being disabled or running in a BIST loopback mode

All other values reserved

SPD The SPD value indicates the negotiated interface communication speed established
- 0000 No negotiated speed (device not present or communication not established)
- 0001 Generation 1 communication rate negotiated

All other values reserved

IPM The IPM value indicates the current interface power management state
- 0000 Device not present or communication not established
- 0001 Interface in active state
- 0010 Interface in PARTIAL power management state
- 0110 Interface in SLUMBER power management state

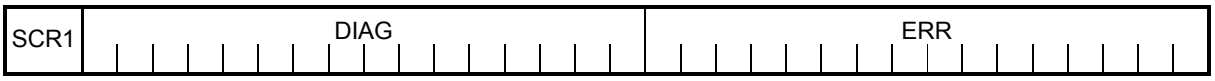
All other values reserved

Reserved All reserved fields shall be cleared to zero.

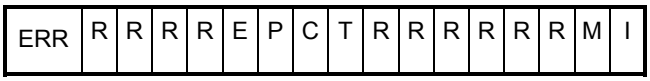
NOTE – The interface must be in the active state for the interface device detection value (DET field) to be accurate. When the interface is in the partial or slumber state no communication between the host and target is established resulting in a DET value corresponding to no device present or no communication established. As a result the insertion or removal of a device may not be accurately detected under all conditions such as when the interface is quiescent as a result of being in sleep or slumber state. This field alone may therefore be insufficient to satisfy all the requirements for device attach or detatch detection during all possible interface states.

10.1.2 SError register

The Serial ATA interface Error register - SError - is a 32-bit register that conveys supplemental Interface error information to complement the error information available in the Shadow Register Block Error register. The register represents all the detected errors accumulated since the last time the SError register was cleared (whether recovered by the interface of not). Set bits in the error register are explicitly cleared by a write operation to the SError register, or a reset operation. The value written to clear set error bits shall have 1's encoded in the bit positions corresponding to the bits that are to be cleared. Host software should clear the Interface SError register at appropriate checkpoints in order to best isolate error conditions and the commands they impact.



ERR The ERR field contains error information for use by host software in determining the appropriate response to the error condition. The field is bit significant as defined in the following figure.



C Non-recovered persistent communication or data integrity error: A communication error that was not recovered occurred that is expected to be persistent. Since the error condition is expected to be persistent the operation need not be retried by host software. Persistent communications errors may arise from faulty interconnect with the device, from a device that has been removed or has failed, or a number of other causes.

E Internal error: The host bus adapter experienced an internal error that caused the operation to fail and may have put the host bus adapter into an error state. Host software should reset the interface before re-trying the operation. If the condition persists, the host bus adapter may suffer from a design issue rendering it incompatible with the attached device.

I Recovered data integrity error: A data integrity error occurred that was recovered by the interface through a retry operation or other recovery action. This can arise from a noise burst in the transmission, a voltage supply variation, or from other causes. No action is required by host software since the operation ultimately succeeded, however, host software may elect to track such recovered errors in order to gauge overall communications integrity and potentially step down the negotiated communication speed.

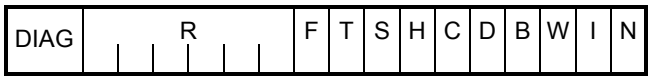
M Recovered communications error: Communications between the device and host was temporarily lost but was re-established. This can arise from a device temporarily being removed, from a temporary loss of Phy synchronization, or from other causes and may be derived from the PhyNRdy signal between the Phy and Link layers. No action is required by the host software since the operation ultimately succeeded, however, host software may elect to track such recovered errors in order to gauge overall communications integrity and potentially step down the negotiated communication speed.

P Protocol error: A violation of the Serial ATA protocol was detected. This can arise from invalid or poorly formed FIS's being received, from invalid state transitions, or from other causes. Host software should reset the interface and retry the corresponding operation. If such an error persists, the attached device may have a design issue rendering it incompatible with the host bus adapter.

R Reserved bit for future use: Shall be cleared to zero.

T Non-recovered transient data integrity error: A data integrity error occurred that was not recovered by the interface. Since the error condition is not expected to be persistent the operation should be retried by host software.

DIAG The DIAG field contains diagnostic error information for use by diagnostic software in validating correct operation or isolating failure modes. The field is bit significant as defined in the following figure.



B 10b to 8b Decode error: When set to a one, this bit indicates that one or more 10b to 8b decoding errors occurred since the bit was last cleared.

C CRC Error: When set to one, this bit indicates that one or more CRC errors occurred with the Link Layer since the bit was last cleared.

D Disparity Error: When set to one, this bit indicates that incorrect disparity was detected one or more times since the last time the bit was cleared.

F Unrecognized FIS type: When set to one, this bit indicates that since the bit was last cleared one or more FIS's were received by the Transport layer with good CRC, but had a type field that was not recognized.

I Phy Internal Error: When set to one, this bit indicates that the Phy detected some internal error since the last time this bit was cleared.

N PhyRdy change: When set to one, this bit indicates that the PhyRdy signal changed state since the last time this bit was cleared.

H Handshake error: When set to one, this bit indicates that one or more R_ERR handshake response was received in response to frame transmission. Such errors may be the result of a CRC error detected by the recipient, a disparity or 10b/8b decoding error, or other error condition leading to a negative handshake on a transmitted frame.

R Reserved bit for future use: Shall be cleared to zero.

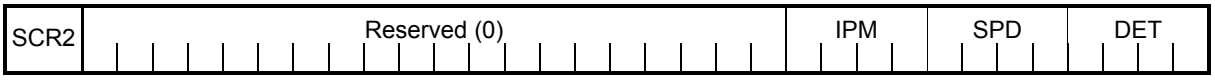
S Link Sequence Error: When set to one, this bit indicates that one or more Link state machine error conditions was encountered since the last time this bit was cleared. The Link Layer state machine defines the conditions under which the link layer detects an erroneous transition.

T Transport state transition error: When set to one, this bit indicates that an error has occurred in the transition from one state to another within the Transport layer since the last time this bit was cleared.

W Comm Wake: When set to one this bit indicates that a Comm Wake signal was detected by the Phy since the last time this bit was cleared.

10.1.3 SControl register

The Serial ATA interface Control register - SControl - is a 32-bit read-write register that provides the interface by which software controls Serial ATA interface capabilities. Writes to the SControl register result in an action being taken by the host adapter or interface. Reads from the register return the last value written to it.



DET The DET field controls the host adapter device detection and interface initialization.
- 0000b No device detection or initialization action requested
- 0001b Perform interface communication initialization sequence to establish communication. This is functionally equivalent to a hard reset and results in the interface being reset and communications reinitialized. Upon a write to the SControl register that sets the LSB of the DET field to one, the host interface shall transition to the HP1: HR_Reset state and shall remain in that state until the LSB of the DET field is cleared to zero by a subsequent write to the SControl register.
- 0100b Disable the Serial ATA interface and put Phy in offline mode.
- All other values reserved

SPD The SPD field represents the highest allowed communication speed the interface is allowed to negotiate when interface communication speed is established
- 0000b No speed negotiation restrictions
- 0001b Limit speed negotiation to a rate not greater than Generation 1 communication rate
- All other values reserved

IPM The IPM field represents the enabled interface power management states that can be invoked via the Serial ATA interface power management capabilities
- 0000b No interface power management state restrictions
- 0001b Transitions to the PARTIAL power management state disabled
- 0010b Transitions to the SLUMBER power management state disabled
- 0011b Transitions to both the PARTIAL and SLUMBER power management states disabled
- All other values reserved

Reserved All reserved fields shall be cleared to zero.

11 Error handling

11.1 Architecture

The layered architecture of Serial ATA extends to error handling as well. As indicated in Figure 67 – Error handling architecture, each layer in the Serial ATA stack has as inputs error indication from the next lower layer (except for the Phy layer which has no lower layer associated with it), data from the next lower layer in the stack, and data from the next higher layer in the stack. Each layer has its local error detection capability to identify errors specific to that layer based on the received data from the lower and higher layers. Each layer performs local recovery and control actions and may forward error information to the next higher layer in the stack.

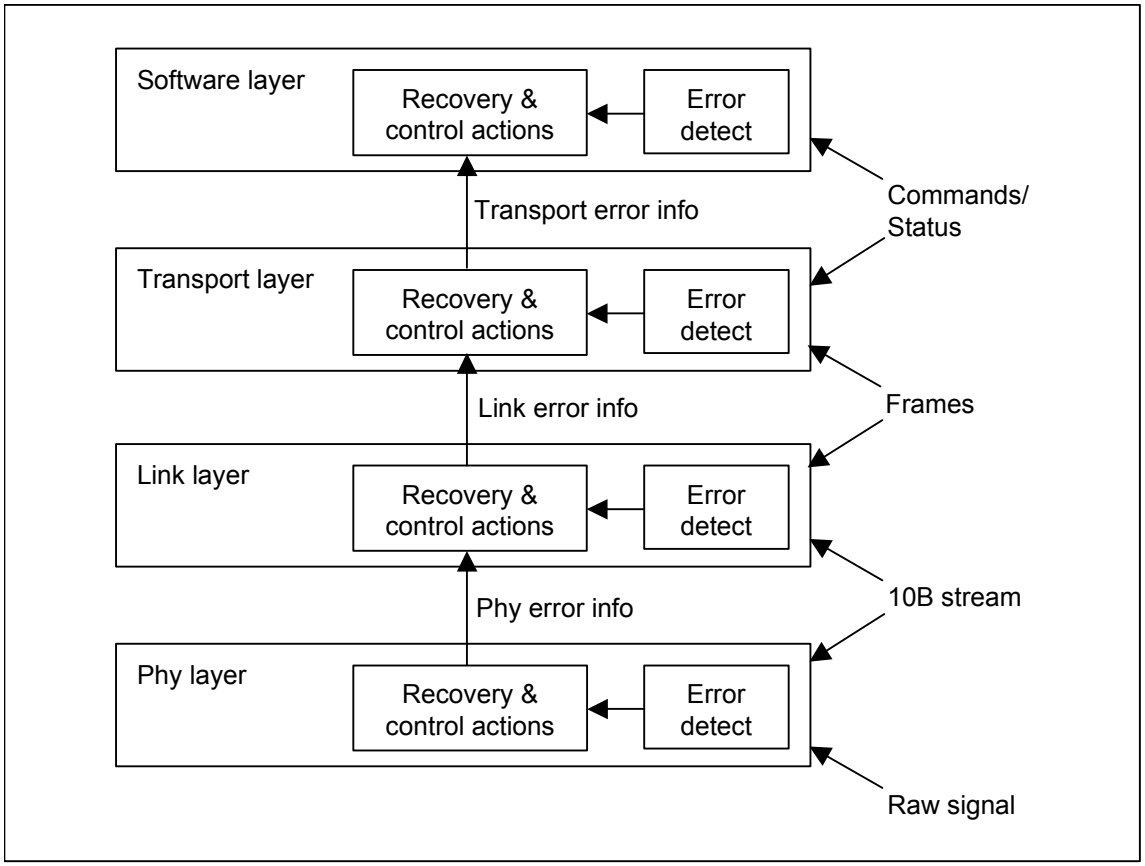


Figure 67 – Error handling architecture

Error responses are generally classified into four categories

Freeze
Abort
Retry
Track/ignore

The error handling responses described in this section are not comprehensive and are included to cover specific known error scenarios as well as to illustrate typical error control and recovery actions. This section is therefore descriptive and supplemental to the error reporting interface defined in section 10 and implementations may vary in their internal error recovery and control actions.

For the most severe error conditions in which state has been critically perturbed in a way that it is not recoverable, the appropriate error response is to freeze and rely on a reset or similar operation to restore all necessary state to return to normal operation.

For error conditions that are expected to be persistent, the appropriate error response is to abort and fail the attempted operation. Such failures usually imply notification up the stack in order to inform host software of the condition.

For error conditions that are transient and not expected to persist, the appropriate response is to retry the failed operation. Only failed operations that have not perturbed system state can be retried. Such retries may either be handled directly by the recovery and error control actions in the relevant layer or may be handled by host software in response to error information being conveyed to it.

Non-critical recoverable conditions can either be tracked or ignored. Such conditions include those that were recovered through a retry or other recovery operation at a lower layer in the stack. Tracking such errors can often be beneficial in identifying a marginally operating component or other imminent failure.

11.2 Phy error handling overview

11.2.1 Error detection

There are three primary categories of error that the Phy layer detects internally:

- No device present
- OOB signaling sequence failure
- Phy internal error (loss of synchronization of communications link)

A no device present condition results from a physical disconnection in the media between the host controller and device, whether intermittent or persistent. The host controller shall detect device presence as part of the interface reset sequence defined in section 6.7 - Functional specifications.

An OOB signaling failure condition arises when the sequence of OOB signaling events cannot be completed, prior to declaration of the "Phy Ready" state. OOB signaling sequences are required when emerging from a power management "partial" or "slumber" state, from a "loopback/BIST" test state, or from initial power-up. OOB signaling sequences are used to achieve a specifically ordered exchange of COMRESET, COMINIT, COMWAKE, and ALIGN patterns to bring the communications link up between host controller and device. The specific sequences are detailed in section 6 - Physical layer.

A Phy Internal Error can arise from a number of conditions, whether it is caused by the characteristics of the input signal, or an internal error unique to the implementation, it will always result in the loss of the synchronization of the communications link.

Fixed local receive PLL frequency architectures (oversampling, tracking/non-tracking) are sensitive to input data rate frequency variations from the nominal expected rate, thus they usually have elasticity buffers. Elasticity buffers are used to accommodate the difference between input data rate frequency, and the local receive PLL frequency -- overrun/underrun conditions will occur if the Tx difference in frequency is too high/low with respect to the Rx local PLL frequency, commonly declared as a Phy Internal Error.

VCO-based (PLL) clock recovery architectures are also sensitive to input data rate frequency variations, high frequency jitter, and may have trouble achieving lock -- which may be declared as a Phy Internal Error.

A number of state machine, impedance compensation, and serializer/deserializer (SerDes) circuits make up a typical Serial ATA interface Phy, and the various types of error conditions can be grouped together to make up the declared "Phy Internal Error". These errors are usually specific to each implementation.

11.2.2 Error control actions

11.2.2.1 No device present

Due to the nature of the physical interface, it is possible for the Phy to determine that a device is attached to the cable at various times. As a direct result, the Phy is responsible for detecting presence of an attached device and this presence shall be reported in the Interface Status register such that host software can respond appropriately.

During the interface initialization sequence, an internal state bit “Device Detect” shall be cleared in the host controller when the COMRESET signal is issued. The “Device Detect” state bit shall be set in the host controller when the host controller detects a COMINIT signal from the attached device. The “Device Detect” state bit corresponds to the device presence detect information in the Interface Status register defined in section 10.1.

Note that device presence and communications established are separately reported in the Interface Status register in order to encompass situations in which an attached device is detected by the Phy, but the Phy is unable to establish communications with it.

11.2.2.2 OOB signaling sequence failure

The Phy does not have any timeout conditions for the interface reset signaling sequence as defined in section 6.7. If a device is present, the Phy shall detect device presence within 10ms of a power-on reset (i.e. COMINIT must be returned within 10ms of an issued COMRESET). If a device is not present, the Phy is not required to time-out and may remain in the reset state indefinitely until host software intervenes. Upon successful completion of the interface initialization sequence, the Phy shall be ready, active, and synchronized, and the Interface Status register bits shall reflect this as defined in section 10.1.

11.2.2.3 Phy internal error

As described in section 11.2.1, there are several potential sources of errors categorized as “Phy Internal Errors.” In order to accommodate a range of implementations without making the software error handling approach implementation dependent, all the different potential sources of internal Phy errors are combined for the purpose of reporting the condition in the Interface Error register as defined in section 10.1.2.. This requirement does not preclude each vendor from implementing their own level of error diagnostic bits, but those shall reside in vendor specific register locations.

Phy internal errors shall result in the Phy becoming not ready (the PhyRdy signal being deasserted) and the corresponding Interface Status register and Interface Error registers bits shall be updated as defined in section 10.1.2.

The "Phy Ready Change" bit, as defined in the Interface Error Register, shall be updated as per its definition in section 10.1.2

11.2.3 Error reporting

Phy errors are generally reported to the Link layer in addition to being reflected in the Interface Status register and Interface Error registers as defined in section 10.1.2.

11.3 Link error handling overview

11.3.1 Error detection

There are two primary categories of errors that the Link layer detects internally:

Invalid state transitions
Data integrity errors

Invalid state transition errors can arise from a number of sources and the Link layer responses to many such error conditions are defined in section 7 - Link layer. Data integrity errors generally arise from noise in the physical interconnect.

11.3.2 Error control actions

Errors detected by the Link during a transmission are generally handled by accumulating the errors until the end of the transmission and reflecting the reception error condition in the final R_ERR/R_OK handshake. Specific scenarios are listed in the following sections.

11.3.2.1 Invalid state transitions

Invalid state transitions are generally handled through the return to a known state, for example where the Link state diagrams in section 7.6 - Link layer state diagrams define the responses for invalid state transition attempts. Returning to a known state is generally achieved through two recovery paths depending on the state of the system.

If the invalid state transitions are attempted during the transmission of a frame (after the receipt of an SOF), the Link shall signal negative acknowledgement (R_ERR) to the transmitting agent.
If the invalid state transition is not during a frame transmission, the Link shall go directly to the idle state, and await the next operation.

The following paragraphs outline the requirements during specific state transition scenarios, and their respective Link error control actions:

Following reception of one or more consecutive X_RDY at the receiver interface, if the next control character received is not SOF, the Link shall notify the Transport Layer of the condition and transition to the idle state.

Following transmission of X_RDY, if there is no returned R_RDY received, no Link recovery action shall be attempted. The higher-level layers will eventually time out, and reset the interface.

On receipt of an unexpected SOF, when the receiving interface had not yet signaled readiness to receive data with the R_RDY signal, that receiving interface shall remain in the idle state issuing

SYNC primitives until the transmitting interface terminates the transmission and also returns to the idle state.

If the transmitter closes a frame with EOF and WTRM, and receives neither a R_OK nor an R_ERR within a predetermined timeout, no Link recovery action shall be attempted. The higher-level layers will eventually time out, and reset the interface.

If the transmitter signals EOF, and a control character other than SYNC, R_OK, or R_ERR is received, the Link layer shall persistently continue to await reception of a proper terminating primitive.

Data integrity errors:
Data integrity errors are generally handled by signaling the Transport Layer in order to potentially trigger a transmission retry operation, or to convey failed status information to the host software. In order to return to a known state, data integrity errors are usually signaled via the frame acknowledgement handshake, at the end of a frame transmission, before returning to the idle state.

The following paragraphs outline the requirements during specific data integrity error scenarios, and the respective Link error control actions:

On detection of a CRC error at the end of receiving a frame (at EOF), the Link Layer shall notify the Transport Layer that the received frame contains a CRC error. Furthermore, the Link Layer shall issue the negative acknowledgement, R_ERR, as the frame status handshake, and shall return to the idle state.

On detection of a disparity error or other 8b/10b coding violation during the receipt of a frame, the Link Layer will retain this error information, and at the close of the received frame the Link Layer shall provide the negative acknowledgement, R_ERR, as the frame handshake, and shall notify the Transport Layer of the error.

The control actions are essentially the same for coding violations as for CRC errors.

11.3.3 Error reporting

Link error conditions are reported to the Transport layer via a private interface between the Link and Transport Layers. Additionally, Link errors are reported in the Interface Error register as defined section 10.1.2.

11.4 Transport error handling overview

The Transport is the highest level layer in the Serial ATA interface. The Transport layer communicates errors to the software and/or performs local error recovery, and initiates control actions, such as retrying a class of FIS transmissions

The Transport layer informs the Link layer of detected errors so that the Link layer can reflect Transport errors in the R_ERR/R_OK handshake at the end of each frame. Devices shall reflect any R_ERR frame handshakes in the command ending status reflected in the transmitted register FIS that conveys the operation ending status. The Transport layer also reflects any encountered error information in the Interface Error register.

The Transport may retry any FIS transmission, provided the system state has not changed as a result of the corresponding failure, and may retry any number of times. For scenarios where repeated retry operations persistently fail, host software will eventually time out the corresponding command and perform recovery operations.

11.4.1 Error detection

In addition to the error information passed to it by the Link layer, the Transport layer internally detects the following categories of errors:

Internal errors
Frame errors
Protocol errors & state errors

There are several kinds of internal errors to the Transport layer, including overflow/underflow of the various speed matching FIFOs. Internal errors are generally handled by failing the corresponding transaction, and returning to a state equivalent to a failed transaction (e. g. the state that would result from a bad CRC).

The Transport layer detects several kinds of frame errors including reception of frames with incorrect CRC, reception of frames with invalid TYPE field, and reception of ill-formed frames (such as a register frames that are not the correct length). Frame errors are generally handled by failing the corresponding transaction and returning to a state equivalent to a failed transaction (such as the state that would result from a bad CRC).

Protocol and state transition errors generally stem from ill-behaved devices not following the proper Serial ATA protocol, and include errors such as, the PIO count value not matching the number of data character subsequently transferred, and errors in the sequence of events.

Protocol and state transition errors are generally handled by failing the corresponding transaction and returning to a state equivalent to a failed transaction (such as the state that would result from a frame being received with a bad CRC).

11.4.2 Error control actions

11.4.2.1 Internal errors

Internal errors are generally handled by failing the corresponding transaction and either re-trying the transaction or notifying host software of the failure condition in order to ultimately generate a host software retry response. The following are specific internal error scenarios and their corresponding Transport error control actions.

If the receive FIFO overflows, the Transport layer shall signal frame reception negative acknowledgement, by signaling the Link layer to return R_ERR during the frame acknowledgement handshake. Subsequent actions are equivalent to a frame reception with erroneous CRC.

If the transmit FIFO underruns, the Transport layer shall close the transmitting frame with an EOF and CRC value that is forced to be incorrect in order to ensure the receiver of the corrupted frame also executes appropriate error control actions. This scenario results only from a transmitter design problem and should never occur for properly implemented devices.

11.4.2.2 Frame errors

Frame errors generally are handled in one of two ways depending on whether the error is expected to be transient or persistent and whether system state has been perturbed. For error conditions expected to be transient (such as a CRC error), and for which the system state has not been perturbed, the Transport layer may retry the corresponding transaction any number of times until ultimately a host timeout and software reset, or other error recovery attempt is made. For error conditions that are not a result of a transient error condition (such as an invalid TYPE field in a received FIS), the error response is generally to fail the transaction and report the failure.

The following are specific frame error scenarios and their corresponding Transport error control actions.

If the Transport receives an FIS with an invalid CRC signaled from the Link layer, the Transport layer shall signal the Link layer to negatively acknowledge frame reception by asserting R_ERR during the frame acknowledgement handshake.

The transmitter of a negatively acknowledged frame may retry the FIS transmission provided the system state has not been perturbed. Frame types that can be retransmitted are:

Register – Host to Device
Register – Device to Host
DMA Activate – Device to Host
DMA Setup – Device to Host
PIO Setup – Device to Host
Set Device Bits – Device to Host

Because data transmission FIS's result in a change in the host bus adapter's internal state, either through the DMA controller changing its state or through a change in the remaining PIO repetition count, data transmission FIS's should never be retried.

The Transport layer is not required to retry those failed FIS transmissions that do not change system state, but the Transport layer may attempt retry any number of times. For conditions that are not addressed through retries, such as persistent errors, host software will eventually time out the transaction and reset the interface.

If the Transport Layer detects reception of an FIS with unrecognized TYPE value, the Transport Layer shall signal the Link Layer to negatively acknowledge the frame reception by asserting R_ERR during the frame acknowledgement handshake.

If the Transport Layer detects reception of a malformed frame, such as a frame with incorrect length, the Transport Layer shall signal the Link Layer to negatively acknowledge the frame reception by asserting R_ERR during the frame acknowledgement handshake.

11.4.2.3 Protocol and state transition errors

Protocol and state errors stem from ill-behaved devices not following the proper protocol. Such errors are generally handled by failing the corresponding transactions and returning to a known state. Since such errors are not caused by an environmental transient, no attempt to retry such failed operations should be made. The following are specific frame error scenarios and their corresponding Transport error control actions.

If the PIO transfer count expires, and two symbols later is not the EOF control character (the CRC falls between the last data character and the EOF), the transfer count stipulated in the PIO Setup FIS did not match the size of the subsequent data payload. For this data-payload/transfer-count mismatch, the Transport Layer shall signal the Link Layer to negatively acknowledge frame reception by asserting R_ERR during the frame acknowledgement handshake.

11.4.3 Error reporting

The Transport Layer reports errors to host software via the Serial ATA Status and Control registers. Devices communicate Transport error information to host software via transmitting a register FIS to update the ATA Shadow Register Block Status and Error register values.

All Transport error conditions that are not handled/recovered by the Transport layer shall set the error bit in the Shadow Register Block Status register, and update the value in the Error register through transmission of an appropriate Register FIS.

Host Transport error conditions shall result in the status and error values in the SStatus and SError register being updated with values corresponding to the error condition and shall result in the Link layer being notified to negatively acknowledge the offending FIS during the final reception handshake.

11.5 Software error handling overview

The software layer error handling is in part defined by legacy software behavior. Superset error reporting capabilities are supported by the Transport layer through the SCR's, and software can take advantage of those error reporting capabilities to improve error handling for Serial ATA.

11.5.1 Error detection

There are three overall error detection mechanisms by which software identifies and responds to Serial ATA errors

- Bad status in the Command Block Status register
- Bad status in the Serial ATA SError register
- Command failed to complete (timeout)

Conditions that return bad status in the Command Block Status register, but for which no Serial ATA interface error information is available, correspond to the error conditions specified in the ATA standard. Such error conditions and responses are defined in the ATA standard and there is no unique handling of those in Serial ATA. Errors in this category include command errors, such as attempts to read from an LBA past the end of the disk, as well as device-specific failures such as data not readable from the given sector number. These failures are not related to the Serial ATA interface, and thus no Serial ATA specific interface status information is available for these error conditions. Only the status information returned by the device is available for identifying the source of the problem, plus any available SMART data that might apply.

Transport layer error conditions, whether recovered or not, are reflected in the Interface Status and Interface Error registers as defined in section 10.1.2. The host Transport layer is responsible for reflecting error information in the Interface Status and Interface Error registers, while the device Transport layer is responsible for reflecting unrecovered errors in the Shadow Register Block Status and Error registers through transmission of appropriate Register FIS's.

Commands that fail to complete are detected by host driver software through a timeout mechanism. Generally such timeouts result in no status or error information for the command being conveyed to host software and software may not be able to determine the source or cause of such errors.

11.5.2 Error control actions

Conditions that return bad status in the Shadow Register Block Status register but for which no interface error information in the SError register is available shall be handled as defined in the ATA standard.

Conditions that return interface error information in the SError register are handled through four basic responses:

Freeze
Abort/Fail
Retry (possible after reset)
Track/ignore

Error conditions that result in catastrophic system perturbation that is not recoverable should result in the system halting. Serial ATA does not define any explicit halting conditions, however, software may be able to infer such conditions.

Error conditions that are not expected to be transient and which are not expected to succeed with subsequent attempts should result in the affected command being aborted and failed. Failure of such commands should be reported to higher software layers for handling. Scenarios in which this response is appropriate include attempts to communicate with a device that is not attached, and failure of the interface to successfully negotiate communications with an attached device.

Error conditions that are expected to be transient should result in the affected command being retried. Such commands may either be retried directly or may be retried after an interface and/or device reset, depending on the particular error value reported in the SError register. Scenarios in which this response is appropriate include noise events resulting in CRC errors, 8b/10b code violations, or disparity errors.

Conditions that are recoverable and for which no explicit error handling is required may be tracked or ignored. Tracking such errors allows subsequent fault isolation for marginal components and accommodates possible recovery operations. Scenarios in which this response is appropriate include tracking the number of Phy synchronization losses in order to identify a potential cable fault or to accommodate an explicit reduction in the negotiated communications rate.

APPENDICES

APPENDIX A. SAMPLE CODE FOR CRC AND SCRAMBLING

A.1 CRC calculation

A.1.1 Overview

The following section provides an informative implementation of the CRC polynomial. The example is intended as an aid in verifying a HDL implementation of the algorithm.

A.1.2 Maximum frame size

The 32-bit CRC used by Serial ATA can be shown to provide detection of two 10-bit errors up to a maximum frame size of 16384 bytes. This will provide for future expansion of Serial ATA FIS's to a maximum of 64 bytes of fixed overhead while still permitting a maximum user data payload of 8192 bytes.

A.1.3 Example code for CRC algorithm

The following code, written in C, illustrates an implementation of the CRC algorithm. A Register Host to Device FIS containing a PIO write command is used as example input. The CRC calculated is the check Dword appended to a transmitted serial stream immediately preceding the EOF primitive.. The code displays both the resulting command FIS and the intermediate CRC polynomial values. The GNU tool chain will compile the code using the following command:

"gcc –o crc.exe crc.c"

This code is supplied for illustrative purposes only.

A.1.4 Example code for CRC algorithm

```
/*****************************************************************************/
/*                                                                           */
/* crc.c                                                                     */
/*                                                                           */
/* This sample code reads standard in for a sequence of 32 bit values       */
/* formatted in hexadecimal with a leading "0x" (e.g. 0xDEADBEEF).  The     */
/* code calculates the Serial ATA CRC for the input data stream.  The       */
/* generator polynomial used is:                                            */
/*         32   26   23   22   16   12   11   10   8   7   5   4   2        */
/* G(x) = x  + x  + x  + x  + x  + x  + x  + x  + x + x + x + x + x + x + 1 */
/*                                                                           */
/* This sample code uses a parallel implementation of the CRC calculation   */
/* circuit that is suitable for implementation in hardware.  A block        */
/* diagram of the circuit being emulated is shown below.                    */
/*                                                                           */
/*                      +---+          +---+          +---+                 */
/*   Data_In ---------->|   |          |   |          | R |                 */
/*                      | + |--------->| * |--------->| e |----+            */
/*               +----->|   |          |   |          | g |    |            */
/*               |      +---+          +---+          +---+    |            */
/*               |                                             |            */
/*               |                                             |            */
/*               +---------------------------------------------+            */
/*                                                                           */
/* The CRC value is initialized to 0x52325032 as defined in the Serial ATA  */
/* specification.                                                            */
/*                                                                           */
/*****************************************************************************/

#include <stdlib.h>
#include <stdio.h>
```

```
main(argc,argv)
int argc;
char *argv[];
{
   int               i,
                     data_count;
   unsigned int      crc,
                     data_in;
   unsigned char     crc_bit[32],
                     new_bit[32];


   crc = 0x52325032;
   data_count = 0;

   while (scanf(" 0x%8x", &data_in) == 1)  {
      data_count++;
      /* Add the data_in value to the current value of the CRC held in the  */
      /* "register".  The addition is performed modulo two (XOR).           */
      crc ^= data_in;
      /* Expand the value of the CRC held in the register to 32 individual  */
      /* bits for easy manipulation.                                        */
      for (i = 0; i < 32; ++i)  {
         crc_bit[i] = (crc >> i) & 0x01;
      }

      /* The following 32 assignments perform the function of the box       */
      /* labeled "*" in the block diagram above.  The new_bit array is a    */
      /* temporary holding place for the new CRC value being calculated.    */
      /* Note that there are lots of shared terms in the assignments below. */
      new_bit[31] = crc_bit[31] ^ crc_bit[30] ^ crc_bit[29] ^ crc_bit[28] ^ crc_bit[27] ^ crc_bit[25] ^ crc_bit[24] ^
                    crc_bit[23] ^ crc_bit[15] ^ crc_bit[11] ^ crc_bit[9]  ^ crc_bit[8]  ^ crc_bit[5];
      new_bit[30] = crc_bit[30] ^ crc_bit[29] ^ crc_bit[28] ^ crc_bit[27] ^ crc_bit[26] ^ crc_bit[24] ^ crc_bit[23] ^
                    crc_bit[22] ^ crc_bit[14] ^ crc_bit[10] ^ crc_bit[8]  ^ crc_bit[7]  ^ crc_bit[4];
      new_bit[29] = crc_bit[31] ^ crc_bit[29] ^ crc_bit[28] ^ crc_bit[27] ^ crc_bit[26] ^ crc_bit[25] ^ crc_bit[23] ^
                    crc_bit[22] ^ crc_bit[21] ^ crc_bit[13] ^ crc_bit[9]  ^ crc_bit[7]  ^ crc_bit[6]  ^ crc_bit[3];
      new_bit[28] = crc_bit[30] ^ crc_bit[28] ^ crc_bit[27] ^ crc_bit[26] ^ crc_bit[25] ^ crc_bit[24] ^ crc_bit[22] ^
                    crc_bit[21] ^ crc_bit[20] ^ crc_bit[12] ^ crc_bit[8]  ^ crc_bit[6]  ^ crc_bit[5]  ^ crc_bit[2];
      new_bit[27] = crc_bit[29] ^ crc_bit[27] ^ crc_bit[26] ^ crc_bit[25] ^ crc_bit[24] ^ crc_bit[23] ^ crc_bit[21] ^
                    crc_bit[20] ^ crc_bit[19] ^ crc_bit[11] ^ crc_bit[7]  ^ crc_bit[5]  ^ crc_bit[4]  ^ crc_bit[1];
      new_bit[26] = crc_bit[31] ^ crc_bit[28] ^ crc_bit[26] ^ crc_bit[25] ^ crc_bit[24] ^ crc_bit[23] ^ crc_bit[22] ^
                    crc_bit[20] ^ crc_bit[19] ^ crc_bit[18] ^ crc_bit[10] ^ crc_bit[6]  ^ crc_bit[4]  ^ crc_bit[3]  ^
                    crc_bit[0];
      new_bit[25] = crc_bit[31] ^ crc_bit[29] ^ crc_bit[28] ^ crc_bit[22] ^ crc_bit[21] ^ crc_bit[19] ^ crc_bit[18] ^
                    crc_bit[17] ^ crc_bit[15] ^ crc_bit[11] ^ crc_bit[8]  ^ crc_bit[3]  ^ crc_bit[2];
      new_bit[24] = crc_bit[30] ^ crc_bit[28] ^ crc_bit[27] ^ crc_bit[21] ^ crc_bit[20] ^ crc_bit[18] ^ crc_bit[17] ^
                    crc_bit[16] ^ crc_bit[14] ^ crc_bit[10] ^ crc_bit[7]  ^ crc_bit[2]  ^ crc_bit[1];
      new_bit[23] = crc_bit[31] ^ crc_bit[29] ^ crc_bit[27] ^ crc_bit[26] ^ crc_bit[20] ^ crc_bit[19] ^ crc_bit[17] ^
                    crc_bit[16] ^ crc_bit[15] ^ crc_bit[13] ^ crc_bit[9]  ^ crc_bit[6]  ^ crc_bit[1]  ^ crc_bit[0];
      new_bit[22] = crc_bit[31] ^ crc_bit[29] ^ crc_bit[27] ^ crc_bit[26] ^ crc_bit[24] ^ crc_bit[23] ^ crc_bit[19] ^
                    crc_bit[18] ^ crc_bit[16] ^ crc_bit[14] ^ crc_bit[12] ^ crc_bit[11] ^ crc_bit[9]  ^ crc_bit[0];
      new_bit[21] = crc_bit[31] ^ crc_bit[29] ^ crc_bit[27] ^ crc_bit[26] ^ crc_bit[24] ^ crc_bit[22] ^ crc_bit[18] ^
                    crc_bit[17] ^ crc_bit[13] ^ crc_bit[10] ^ crc_bit[9]  ^ crc_bit[5];
      new_bit[20] = crc_bit[30] ^ crc_bit[28] ^ crc_bit[26] ^ crc_bit[25] ^ crc_bit[23] ^ crc_bit[21] ^ crc_bit[17] ^
                    crc_bit[16] ^ crc_bit[12] ^ crc_bit[9]  ^ crc_bit[8]  ^ crc_bit[4];
      new_bit[19] = crc_bit[29] ^ crc_bit[27] ^ crc_bit[25] ^ crc_bit[24] ^ crc_bit[22] ^ crc_bit[20] ^ crc_bit[16] ^
                    crc_bit[15] ^ crc_bit[11] ^ crc_bit[8]  ^ crc_bit[7]  ^ crc_bit[3];
      new_bit[18] = crc_bit[31] ^ crc_bit[28] ^ crc_bit[26] ^ crc_bit[24] ^ crc_bit[23] ^ crc_bit[21] ^ crc_bit[19] ^
                    crc_bit[15] ^ crc_bit[14] ^ crc_bit[10] ^ crc_bit[7]  ^ crc_bit[6]  ^ crc_bit[2];
      new_bit[17] = crc_bit[31] ^ crc_bit[30] ^ crc_bit[27] ^ crc_bit[25] ^ crc_bit[23] ^ crc_bit[22] ^ crc_bit[20] ^
                    crc_bit[18] ^ crc_bit[14] ^ crc_bit[13] ^ crc_bit[9]  ^ crc_bit[6]  ^ crc_bit[5]  ^ crc_bit[1];
      new_bit[16] = crc_bit[30] ^ crc_bit[29] ^ crc_bit[26] ^ crc_bit[24] ^ crc_bit[22] ^ crc_bit[21] ^ crc_bit[19] ^
                    crc_bit[17] ^ crc_bit[13] ^ crc_bit[12] ^ crc_bit[8]  ^ crc_bit[5]  ^ crc_bit[4]  ^ crc_bit[0];
      new_bit[15] = crc_bit[30] ^ crc_bit[27] ^ crc_bit[24] ^ crc_bit[21] ^ crc_bit[20] ^ crc_bit[18] ^ crc_bit[16] ^
                    crc_bit[15] ^ crc_bit[12] ^ crc_bit[9]  ^ crc_bit[8]  ^ crc_bit[7]  ^ crc_bit[5]  ^ crc_bit[4]  ^
                    crc_bit[3];
      new_bit[14] = crc_bit[29] ^ crc_bit[26] ^ crc_bit[23] ^ crc_bit[20] ^ crc_bit[19] ^ crc_bit[17] ^ crc_bit[15] ^
                    crc_bit[14] ^ crc_bit[11] ^ crc_bit[8]  ^ crc_bit[7]  ^ crc_bit[6]  ^ crc_bit[4]  ^ crc_bit[3]  ^
                    crc_bit[2];
      new_bit[13] = crc_bit[31] ^ crc_bit[28] ^ crc_bit[25] ^ crc_bit[22] ^ crc_bit[19] ^ crc_bit[18] ^ crc_bit[16] ^
                    crc_bit[14] ^ crc_bit[13] ^ crc_bit[10] ^ crc_bit[7]  ^ crc_bit[6]  ^ crc_bit[5]  ^ crc_bit[3]  ^
                    crc_bit[2]  ^ crc_bit[1];
      new_bit[12] = crc_bit[31] ^ crc_bit[30] ^ crc_bit[27] ^ crc_bit[24] ^ crc_bit[21] ^ crc_bit[18] ^ crc_bit[17] ^
                    crc_bit[15] ^ crc_bit[13] ^ crc_bit[12] ^ crc_bit[9]  ^ crc_bit[6]  ^ crc_bit[5]  ^ crc_bit[4]  ^
                    crc_bit[2]  ^ crc_bit[1]  ^ crc_bit[0];
      new_bit[11] = crc_bit[31] ^ crc_bit[28] ^ crc_bit[27] ^ crc_bit[26] ^ crc_bit[25] ^ crc_bit[24] ^ crc_bit[20] ^
                    crc_bit[17] ^ crc_bit[16] ^ crc_bit[15] ^ crc_bit[14] ^ crc_bit[12] ^ crc_bit[9]  ^ crc_bit[4]  ^
                    crc_bit[3]  ^ crc_bit[1]  ^ crc_bit[0];
      new_bit[10] = crc_bit[31] ^ crc_bit[29] ^ crc_bit[28] ^ crc_bit[26] ^ crc_bit[19] ^ crc_bit[16] ^ crc_bit[14] ^
                    crc_bit[13] ^ crc_bit[9]  ^ crc_bit[5]  ^ crc_bit[3]  ^ crc_bit[2]  ^ crc_bit[0];
      new_bit[9]  = crc_bit[29] ^ crc_bit[24] ^ crc_bit[23] ^ crc_bit[18] ^ crc_bit[13] ^ crc_bit[12] ^ crc_bit[11] ^
```

```
                crc_bit[9]  ^ crc_bit[5]  ^ crc_bit[4]  ^ crc_bit[2]  ^ crc_bit[1];
    new_bit[8]  = crc_bit[31] ^ crc_bit[28] ^ crc_bit[23] ^ crc_bit[22] ^ crc_bit[17] ^ crc_bit[12] ^ crc_bit[11] ^
                crc_bit[10] ^ crc_bit[8]  ^ crc_bit[4]  ^ crc_bit[3]  ^ crc_bit[1]  ^ crc_bit[0];
    new_bit[7]  = crc_bit[29] ^ crc_bit[28] ^ crc_bit[25] ^ crc_bit[24] ^ crc_bit[23] ^ crc_bit[22] ^ crc_bit[21] ^
                crc_bit[16] ^ crc_bit[15] ^ crc_bit[10] ^ crc_bit[8]  ^ crc_bit[7]  ^ crc_bit[5]  ^ crc_bit[3]  ^
                crc_bit[2]  ^ crc_bit[0];
    new_bit[6]  = crc_bit[30] ^ crc_bit[29] ^ crc_bit[25] ^ crc_bit[22] ^ crc_bit[21] ^ crc_bit[20] ^ crc_bit[14] ^
                crc_bit[11] ^ crc_bit[8]  ^ crc_bit[7]  ^ crc_bit[6]  ^ crc_bit[5]  ^ crc_bit[4]  ^ crc_bit[2]  ^
                crc_bit[1];
    new_bit[5]  = crc_bit[29] ^ crc_bit[28] ^ crc_bit[24] ^ crc_bit[21] ^ crc_bit[20] ^ crc_bit[19] ^ crc_bit[13] ^
                crc_bit[10] ^ crc_bit[7]  ^ crc_bit[6]  ^ crc_bit[5]  ^ crc_bit[4]  ^ crc_bit[3]  ^ crc_bit[1]  ^
                crc_bit[0];
    new_bit[4]  = crc_bit[31] ^ crc_bit[30] ^ crc_bit[29] ^ crc_bit[25] ^ crc_bit[24] ^ crc_bit[20] ^ crc_bit[19] ^
                crc_bit[18] ^ crc_bit[15] ^ crc_bit[12] ^ crc_bit[11] ^ crc_bit[8]  ^ crc_bit[6]  ^ crc_bit[4]  ^
                crc_bit[3]  ^ crc_bit[2]  ^ crc_bit[0];
    new_bit[3]  = crc_bit[31] ^ crc_bit[27] ^ crc_bit[25] ^ crc_bit[19] ^ crc_bit[18] ^ crc_bit[17] ^ crc_bit[15] ^
                crc_bit[14] ^ crc_bit[10] ^ crc_bit[9]  ^ crc_bit[8]  ^ crc_bit[7]  ^ crc_bit[3]  ^ crc_bit[2]  ^
                crc_bit[1];
    new_bit[2]  = crc_bit[31] ^ crc_bit[30] ^ crc_bit[26] ^ crc_bit[24] ^ crc_bit[18] ^ crc_bit[17] ^ crc_bit[16] ^
                crc_bit[14] ^ crc_bit[13] ^ crc_bit[9]  ^ crc_bit[8]  ^ crc_bit[7]  ^ crc_bit[6]  ^ crc_bit[2]  ^
                crc_bit[1]  ^ crc_bit[0];
    new_bit[1]  = crc_bit[28] ^ crc_bit[27] ^ crc_bit[24] ^ crc_bit[17] ^ crc_bit[16] ^ crc_bit[13] ^ crc_bit[12] ^
                crc_bit[11] ^ crc_bit[9]  ^ crc_bit[7]  ^ crc_bit[6]  ^ crc_bit[1]  ^ crc_bit[0];
    new_bit[0]  = crc_bit[31] ^ crc_bit[30] ^ crc_bit[29] ^ crc_bit[28] ^ crc_bit[26] ^ crc_bit[25] ^ crc_bit[24] ^
                crc_bit[16] ^ crc_bit[12] ^ crc_bit[10] ^ crc_bit[9]  ^ crc_bit[6]  ^ crc_bit[0];


    /* The new CRC value has been calculated as individual bits in the    */
    /* new_bit array.  Re-assembled it into a 32 bit value and "clock" it */
    /* into the "register".                                               */
    crc = 0;
    for (i = 31; i >= 0; --i)  {
       crc = crc << 1;
       crc |= new_bit[i];
    }
    printf("Running CRC value is 0x%08X\n", crc);
  }

  printf("\n\nThe total number of data words processed was %d\n", data_count);
  printf("The CRC is 0x%08X\n\n", crc);

  return 0;
}
```

A.1.5 Example CRC implementation output

The following is the sample data used as input for the example stored in file *sample*:

```
0x00308027
0xE1234567
0x00000000
0x00000002
0x00000000
```

Executing the command *./crc* < *sample* yields the following output:

```
Running CRC value is 0x11E353FD
Running CRC value is 0x0F656DA7
Running CRC value is 0x3D14369C
Running CRC value is 0x92D0D681
Running CRC value is 0x319FFF6F


The total number of data words processed was 5
The CRC is 0x319FFF6F
```

A.2 Scrambling calculation

A.2.1 Overview

The following section provides an informative implementation of the scrambling polynomial. The example is intended as an aid in verifying a HDL implementation of the algorithm.

A.2.2 Example code for scrambling algorithm

The following code, written in C, illustrates an implementation of the Scrambling algorithm. A Register Host to Device FIS containing a PIO write command is used as example input. The code displays both the resulting scrambled data and the raw output of the scrambler. The GNU tool chain will compile the code using the following command:

"gcc –o scramble.exe scramble.c"

This code is supplied for illustrative purposes only.

A.2.3 Example scrambler implementation

```
/*************************************************************************/
/*                                                                       */
/* scramble.c                                                            */
/*                                                                       */
/* This sample code generates the entire sequence of 65535 Dwords produced  */
/* by the scrambler defined in the Serial ATA specification.  The        */
/* specification calls for an LFSR to generate a string of bits that will   */
/* be packaged into 32 bit Dwords to be XORed with the data Dwords.  The    */
/* generator polynomial specified is:                                    */
/*                          16   15   13   4                             */
/*                 G(x) = x  + x  + x  + x + 1                           */
/*                                                                       */
/* Parallelized versions of the scrambler are initialized to a value     */
/* derived from the initialization value of 0xFFFF defined in the        */
/* specification.  This implementation is initialized to 0xF0F6.  Other  */
/* parallel implementations will have different initial values.  The     */
/* important point is that the first Dword output of any implementation  */
/* must equal 0xC2D2768D.                                                */
/*                                                                       */
/* This code does not represent an elegant solution for a C implementation, */
/* but it does demonstrate a method of generating the sequence that can be  */
/* easily implemented in hardware.  A block diagram of the circuit emulated */
/* by this code is shown below.                                          */
/*                                                                       */
/*         +-----------------------------------+                         */
/*         |                                   |                         */
/*         |                                   |                         */
/*         |     +---+               +---+     |                         */
/*         |     | R |               | * |     |                         */
/*         +---->| e |---------+---->| M |---+---> Output(31 downto 16)  */
/*               | g |         |     | 1 |                               */
/*               +---+         |     +---+                               */
/*                             |                                         */
/*                             |     +---+                               */
/*                             |     | * |                               */
/*                             +---->| M |---------> Output(15 downto 0)   */
/*                                   | 2 |                               */
/*                                   +---+                               */
/*                                                                       */
/* The register shown in the block diagram is a 16 bit register.  The two   */
/* boxes, *M1 and *M2, each represent a multiply by a 16 by 16 binary    */
/* matrix.  A 16 by 16 matrix times a 16 bit vector yields a 16 bit vector. */
/* The two vectors are the two halves of the 32 bit scrambler value.  The   */
/* upper half of the scrambler value is stored back into the context     */
/* register to be used to generate the next value in the scrambler       */
/* sequence.                                                             */
/*                                                                       */
/*************************************************************************/
```

```
#include <stdlib.h>
#include <stdio.h>


main(argc,argv)
int argc;
char *argv[];


{
   int               i, j;
   unsigned short    context;        /* The 16 bit register that holds the context or state */
   unsigned long     scrambler;      /* The 32 bit output of the circuit    */
   unsigned char     now[16];        /* The individual bits of context      */
   unsigned char     next[32];       /* The computed bits of scrambler      */



   /* Parallelized versions of the scrambler are initialized to a value     */
   /* derived from the initialization value of 0xFFFF defined in the        */
   /* specification.  This implementation is initialized to 0xF0F6.  Other  */
   /* parallel implementations will have different initial values.  The     */
   /* important point is that the first Dword output of any implementation  */
   /* must equal 0xC2D2768D.                                                */
   context = 0xF0F6;


   for (i = 0; i < 65535; ++i)  {
      /* Split the register contents (the variable context) up into its     */
      /* individual bits for easy handling.                                 */
      for (j = 0; j < 16; ++j)  {
         now[j] = (context >> j) & 0x01;
      }

      /* The following 16 assignments implement the matrix multiplication   */
      /* performed by the box labeled *M1.                                  */
      /* Notice that there are lots of shared terms in these assignments.   */
      next[31] = now[12] ^ now[10] ^ now[7]  ^ now[3]  ^ now[1]  ^ now[0];
      next[30] = now[15] ^ now[14] ^ now[12] ^ now[11] ^ now[9]  ^ now[6]  ^ now[3]  ^ now[2]  ^ now[0];
      next[29] = now[15] ^ now[13] ^ now[12] ^ now[11] ^ now[10] ^ now[8]  ^ now[5]  ^ now[3]  ^ now[2]  ^ now[1];
      next[28] = now[14] ^ now[12] ^ now[11] ^ now[10] ^ now[9]  ^ now[7]  ^ now[4]  ^ now[2]  ^ now[1]  ^ now[0];
      next[27] = now[15] ^ now[14] ^ now[13] ^ now[12] ^ now[11] ^ now[10] ^ now[9]  ^ now[8]  ^ now[6]  ^ now[1]  ^ now[0];
      next[26] = now[15] ^ now[13] ^ now[11] ^ now[10] ^ now[9]  ^ now[8]  ^ now[7]  ^ now[5]  ^ now[3]  ^ now[0];
      next[25] = now[15] ^ now[10] ^ now[9]  ^ now[8]  ^ now[7]  ^ now[6]  ^ now[4]  ^ now[3]  ^ now[2];
      next[24] = now[14] ^ now[9]  ^ now[8]  ^ now[7]  ^ now[6]  ^ now[5]  ^ now[3]  ^ now[2]  ^ now[1];
      next[23] = now[13] ^ now[8]  ^ now[7]  ^ now[6]  ^ now[5]  ^ now[4]  ^ now[2]  ^ now[1]  ^ now[0];
      next[22] = now[15] ^ now[14] ^ now[7]  ^ now[6]  ^ now[5]  ^ now[4]  ^ now[1]  ^ now[0];
      next[21] = now[15] ^ now[13] ^ now[12] ^ now[6]  ^ now[5]  ^ now[4]  ^ now[0];
      next[20] = now[15] ^ now[11] ^ now[5]  ^ now[4];
      next[19] = now[14] ^ now[10] ^ now[4]  ^ now[3];
      next[18] = now[13] ^ now[9]  ^ now[3]  ^ now[2];
      next[17] = now[12] ^ now[8]  ^ now[2]  ^ now[1];
      next[16] = now[11] ^ now[7]  ^ now[1]  ^ now[0];

      /* The following 16 assignments implement the matrix multiplication   */
      /* performed by the box labeled *M2.                                  */
      next[15] = now[15] ^ now[14] ^ now[12] ^ now[10] ^ now[6]  ^ now[3]  ^ now[0];
      next[14] = now[15] ^ now[13] ^ now[12] ^ now[11] ^ now[9]  ^ now[5]  ^ now[3]  ^ now[2];
      next[13] = now[14] ^ now[12] ^ now[11] ^ now[10] ^ now[8]  ^ now[4]  ^ now[2]  ^ now[1];
      next[12] = now[13] ^ now[11] ^ now[10] ^ now[9]  ^ now[7]  ^ now[3]  ^ now[1]  ^ now[0];
      next[11] = now[15] ^ now[14] ^ now[10] ^ now[9]  ^ now[8]  ^ now[6]  ^ now[3]  ^ now[2]  ^ now[0];
      next[10] = now[15] ^ now[13] ^ now[12] ^ now[9]  ^ now[8]  ^ now[7]  ^ now[5]  ^ now[3]  ^ now[2]  ^ now[1];
      next[9]  = now[14] ^ now[12] ^ now[11] ^ now[8]  ^ now[7]  ^ now[6]  ^ now[4]  ^ now[2]  ^ now[1]  ^ now[0];
      next[8]  = now[15] ^ now[14] ^ now[13] ^ now[12] ^ now[11] ^ now[10] ^ now[7]  ^ now[6]  ^ now[5]  ^ now[1]  ^ now[0];
      next[7]  = now[15] ^ now[13] ^ now[11] ^ now[10] ^ now[9]  ^ now[6]  ^ now[5]  ^ now[4]  ^ now[3]  ^ now[0];
      next[6]  = now[15] ^ now[10] ^ now[9]  ^ now[8]  ^ now[5]  ^ now[4]  ^ now[2];
      next[5]  = now[14] ^ now[9]  ^ now[8]  ^ now[7]  ^ now[4]  ^ now[3]  ^ now[1];
      next[4]  = now[13] ^ now[8]  ^ now[7]  ^ now[6]  ^ now[3]  ^ now[2]  ^ now[0];
      next[3]  = now[15] ^ now[14] ^ now[7]  ^ now[6]  ^ now[5]  ^ now[3]  ^ now[2]  ^ now[1];
      next[2]  = now[14] ^ now[13] ^ now[6]  ^ now[5]  ^ now[4]  ^ now[2]  ^ now[1]  ^ now[0];
      next[1]  = now[15] ^ now[14] ^ now[13] ^ now[5]  ^ now[4]  ^ now[1]  ^ now[0];
      next[0]  = now[15] ^ now[13] ^ now[4]  ^ now[0];

      /* The 32 bits of the output have been generated in the "next" array. */
      /* Reassemble the bits into a 32 bit Dword.                           */
      scrambler = 0;
      for (j = 31; j >= 0; --j)  {
         scrambler = scrambler << 1;
         scrambler |= next[j];
      }
      /* The upper half of the scrambler output is stored backed into the   */
```

```
      /* register as the saved context for the next cycle.                */
      context = scrambler >> 16;
      printf("0x%08X\n", scrambler);

   }

   return 0;

}
```

A.2.4 Example scrambler implementation output

The following lists the first 32 results generated by the scrambler and the C sample code listed above.

```
0xC2D2768D
0x1F26B368
0xA508436C
0x3452D354
0x8A559502
0xBB1ABE1B
0xFA56B73D
0x53F60B1B
0xF0809C41
0x747FC34A
0xBE865291
0x7A6FA7B6
0x3163E6D6
0xF036FE0C
0x1EF3EA29
0xEB342694
0x53853B17
0xE94ADC4D
0x5D200E88
0x6901EDD0
0xFA9E38DE
0x68DB4B07
0x450A437B
0x960DD708
0x3F35E698
0xFE7698A5
0xC80EF715
0x666090AF
0xFAF0D5CB
0x2B82009F
0x0E317491
0x76F46A1E
```

A.3 Example frame

The following table shows the steps and values used in the transmission of a simple FIS.

For LBA = 1234567 and Sector Count = 2

Table 29. – CRC and scrambler calculation example - PIO Write Command

FIS Data	Scrambler Value	Scrambled Data	Accumulated CRC Value	Comments
3737B57C	N/A	3737B57C	N/A	SOF, primitives not scrambled, scrambler reset
00308027	C2D2768D	C2E2F6AA	11E353FD	RegH2D, Command = 30, Cbit set
E1234567	1F26B368	FE05F60F	0F656DA7	LBA 1234567
00000000	A508436C	A508436C	3D14369C	Extended LBA = 0
00000002	3452D354	3452D356	92D0D681	Control Reg = 0, 2 sectors
00000000	8A559502	8A559502	319FFF6F	reserved = 0
319FFF6F	BB1ABE1B	8A854174	N/A	CRC
D5D5B57C	N/A	D5D5B57C	N/A	EOF, primitives not scrambled

Note: All values in hexadecimal, shown 31:0

The transmitted dwords for this register FIS, prior to 8b/10b encoding, are :

SOF
c2e2f6aa
fe05f60f
a508436c
3452d356
8a559502
8A854174
EOF

APPENDIX B. Type field value selection

The values for the TYPE field of the FIS's have been selected in order to provide additional robustness. In minimally buffered implementations that may not buffer a complete FIS, the state machines may begin acting on the received FIS TYPE value prior to the ending CRC having been checked. Because the TYPE value may be acted upon prior to the integrity of the complete FIS being checked against its ending CRC, the TYPE field values have been selected to maximize the Hamming distance between them.

Several considerations are made in selecting the TYPE field values. Since the value is 8b/10b encoded, the Hamming distance for the values after 10b encoding is maximized. Since the starting running disparity at the time the TYPE field is transmitted is not known, values have been selected that have the same encoding for both negative and positive starting running disparity.

B.1 Type field values

Table 30 lists the selected TYPE field values (with their 10b encoding) that have a Hamming distance of 4 or greater in the 10b space after scrambling and encoding and which have the same encoding for both positive and negative running disparity.

Table 30. – Type field values

Type field value (hex)	Scrambler syndrome[1] (hex)	8b scrambled value (hex)	10b encoding (binary)
0x27	0x8D	0xAA	0101011010
0x34	0x8D	0xB9	1001101010
0x39	0x8D	0xB4	0010111010
0x41	0x8D	0xCC	0011010110
0x46	0x8D	0xCB	1101000110
0x58	0x8D	0xD5	1010100110
0x5F	0x8D	0xD2	0100110110
0xA1	0x8D	0x2C	0011011001
0xA6	0x8D	0x2B	1101001001
0xB8	0x8D	0x35	1010101001
0xBF	0x8D	0x32	0100111001
0xC7	0x8D	0x4A	0101010101
0xD4	0x8D	0x59	1001100101
0xD9	0x8D	0x54	0010110101

NOTE –
1. The scrambler syndrome generator is reset on the SOF primitive and the type field value immediately follows the SOF primitive. The scrambler syndrome at the time the type field value is transmitted is therefore deterministic and is equal to the seed used for the scrambling generator. See the scrambler generator section for more information on the details of the scrambler and its generated syndromes.

APPENDIX C. Further information about cable and connector

C.1 Device connector configurations

Serial ATA defines several device connector configurations, as described below.

C.1.1 Configuration 35A1

Configuration 35A1 is the configuration for 3.5" form factor devices without any legacy connector support.

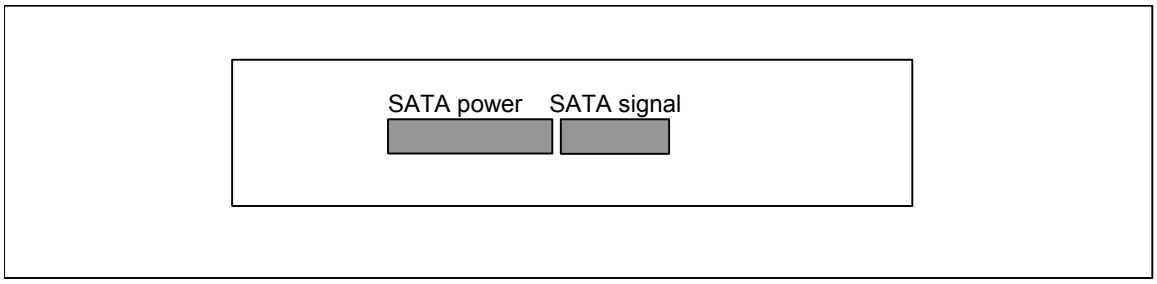


Figure 68 - Device connector configuration 35A1

C.1.2 Configuration 35B1

Configuration 35B1 includes Serial ATA signal and power connectors and legacy jumper and power connectors.

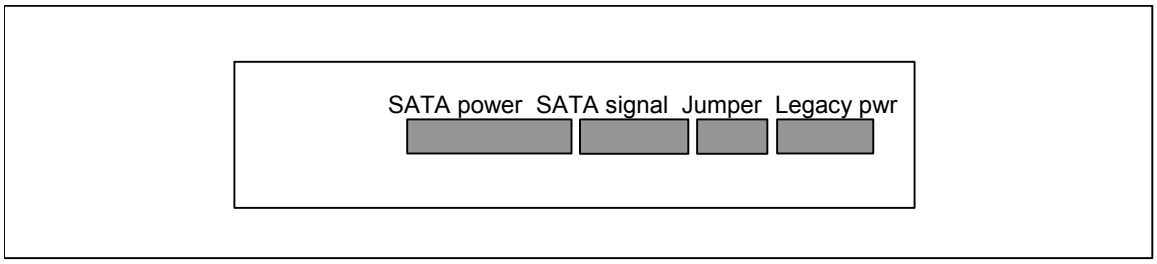


Figure 69 - Device connector configuration 35B1

Configuration 35B2

Configuration 35B2 includes a Serial ATA signal connector and legacy jumper and power connectors.

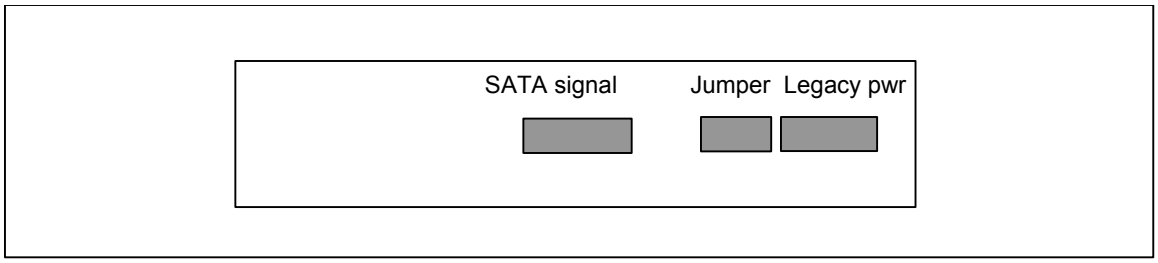


Figure 70 - Device connector configuration 35B2

C.1.3 Configuration 35B3

Configuration 35B3 includes a Serial ATA signal connector and a legacy power connector.

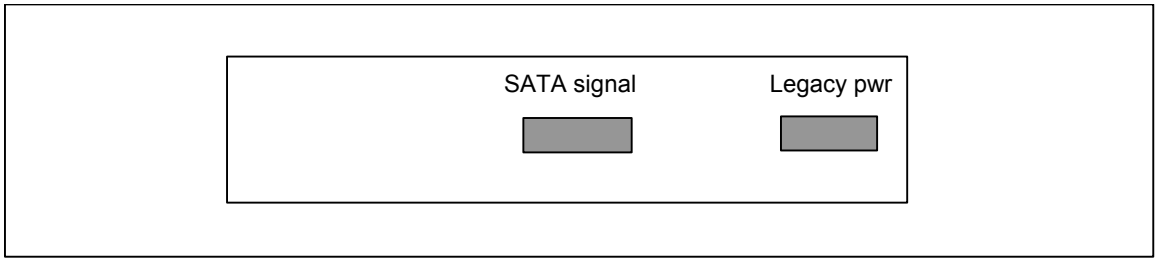


Figure 71 - Device connector configuration 35B3

C.1.4 Configuration 35B4

Configuration 35B4 includes Serial ATA signal and power connectors and a legacy jumper connector.

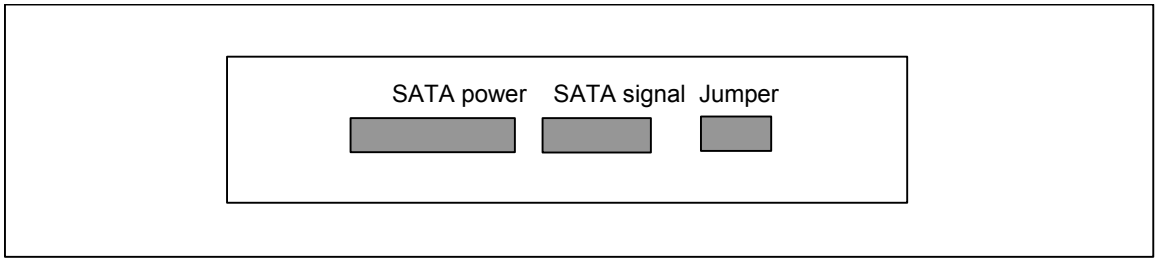


Figure 72 - Device connector configuration 35B4

C.1.5 Configuration 35B5

Configuration 35B5 includes Serial ATA signal and power connectors and a legacy power connector.

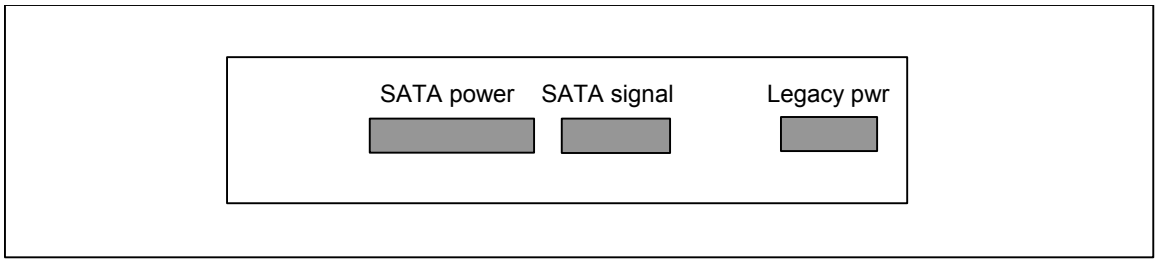


Figure 73 - Device connector configuration 35B5

C.1.6 Configuration 35C1

Configuration 35C1 includes a legacy Parallel ATA signal connector, a Serial ATA signal connector and a legacy power connector.

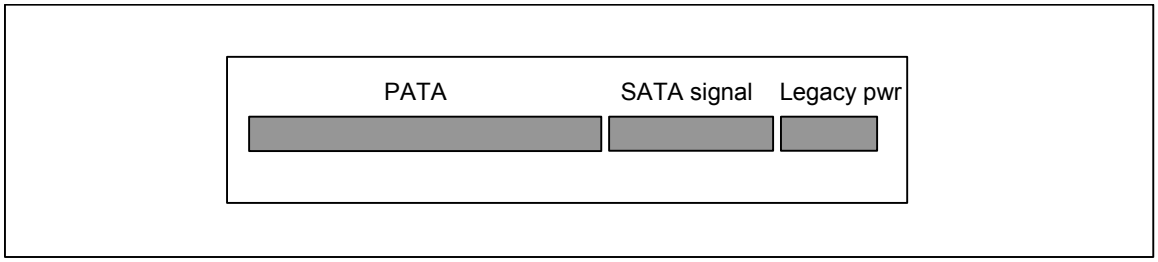


Figure 74 - Device connector configuration 35C1

The presence of the parallel ATA connector prevents the same connector location as for the other 3.5” device configurations.

C.1.7 Configuration 25A1

Configuration 25A1 is the only configuration allowed for 2.5" form factor devices.

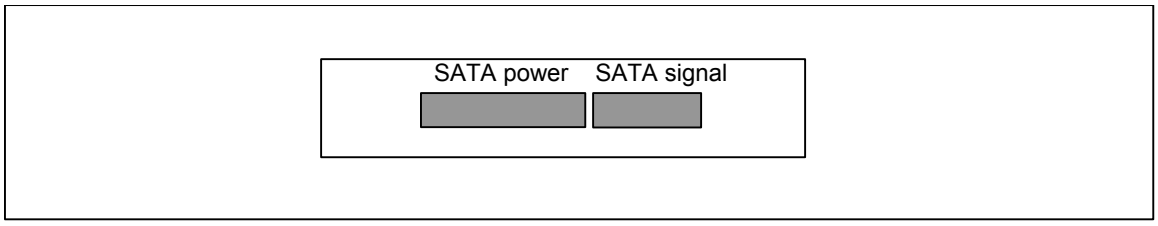
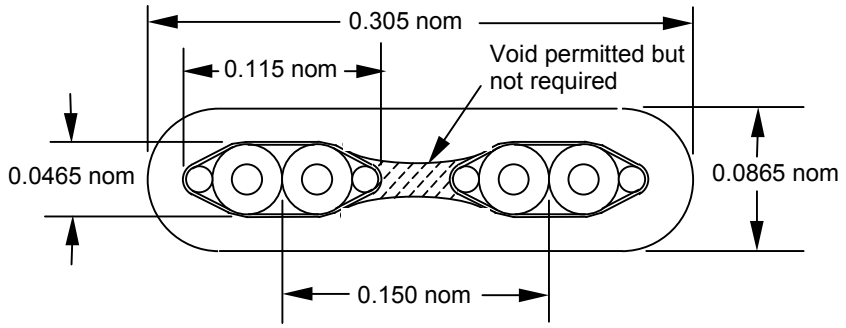


Figure 75 - Device connector configuration 25A1

C.2 Cable construction example

Figure 76 shows a Serial ATA cable construction example.



Electrical
- Impedance: 100 ± 5 Ω (differential)
- Capacitance: 42 pf/M nominal
- Prop delay: 4.25 ns/M nominal
- Skew (within pair): TBD (TDT method, drains gounded) (differential 50%-50%, Tektronix 11801, SD-24/SD-26 sampling heads)
- Attenuation (nominal): TBD db/M @ 4.5 GHz

Physical
- (2) shielded pairs
  - 26 AWG solid tinned copper
  - 0.0435 nom diameter foam polyolefin, white
  - Parallel pair with (2) drains, 28 AWG solid tinned copper
  - 0.001aluminized polyester, foil in, 0.035 inch min overlap
  - Blue typical longitudinal wrap, heat sealed
- Jacket: 0.020 nom wall PVC, red

All dimensions are inches unless otherwise noted

Figure 76 – Cable construction example

C.3 Contact material and plating

Table 31 shows the recommendations for the contact material and plating.

Table 31 – Contact material and plating recommendations

Parameter	Recommendation	Comments
Material	Copper alloys, for example, brass for plug contacts and phosphor bronze for receptacle contact, the spring.	Material temper and thickness should be selected based on normal force and elastic deflection range consideration.
Mating Side Plating	For 50 durability cycles: • 1.27 μm (50 μin) minimum Ni with either 0.38 μm (15 μin) minimum Au or 0.38 μm (15 μin) minimum 80/20 Pd/Ni with 0.051 μm (2 μin) minimum Au flash For 500 durability cycles: • 1.27 μm (50 μin) minimum Ni with either 0.76 μm (30 μin) minimum Au or 0.76 μm (30 μin) minimum 80/20 Pd/Ni with 0.051 μm (2 μin) minimum Au flash	Exposed underplate or base material is not allowed in the mating area.
Solder Side Plating	Either Sn/Pb plating or Pd/Ni with Au flash: • 1.27 μm (50 μin) minimum Ni with 3.18 μm (125 μin) minimum Sn/Pb. Or • 1.27 μm (50 μin) minimum Ni with 0.76 μm (30 μin) minimum 80/20 Pd/Ni with 0.051 μm (2 μin) minimum Au flash.	Exposed base material is allowed in small areas where the contact is excised from its carrier strip or bandolier.

Appendix D. Command processing overview

D.1 Non-data commands

When the Command register is written by the BIOS or software driver, the host adapter sets BSY in the shadow Status register and transmits a Command frame to the device.

When command actions are complete, the device transmits a Register frame to set ending content of the shadow registers

D.1.1 Legacy DMA read by host from device

- Prior to the command being issued to the device, the host driver software programs the host adapter's DMA controller with the memory address pointer(s) and the transfer direction, and arms the DMA controller (enables the "run" flag).
- The host driver software issues the command to the device by writing the Shadow Register Block Registers (command register last).
- In response to the command Shadow Register Block Register being written, the host adapter sets the BSY bit in the status shadow register and transmits a Register – Host to Device FIS to the device with the Shadow Register Block contents.
- When the device has processed the command and is ready, it transmits the read data to the host in the form of one or more Data FIS's. This transfer proceeds in response to flow control signals/readiness.
- The host adapter recognizes that the incoming frame is a Data FIS and the DMA controller is programmed, and directs the incoming data to the host adapter's DMA controller which forwards the incoming data to the appropriate host memory locations.
- Upon completion of the transfer, the device transmits a Register – Device to Host FIS to indicate ending status for the command, clearing the BSY bit in the Status register, and if the interrupt flag is set in the header an interrupt is asserted to the host.

In some error conditions there may be no Data FIS transmitted prior to the Register FIS being transmitted with the status information. If so, the host software driver shall abort the setup of the DMA controller.

D.1.2 Legacy DMA write by host to device

- Prior to the command being issued to the device, the host driver software programs the host adapter's DMA controller with the memory address pointer(s) and the transfer direction, and arms the DMA controller (enables the "run" flag). As a result the DMA controller becomes armed but remains paused pending a signal from the device to proceed with the data transfer.
- The host driver software issues the command to the device by writing the Shadow Register Block Registers (command register last).
- In response to the command Shadow Register Block Register being written, the host adapter sets the BSY bit in the status shadow register and transmits a Register – Host to Device FIS to the device with the Shadow Register Block contents.
- When the device is ready to receive the data from the host, the device transmits a DMA Activate FIS to the host, which activates the armed DMA controller. The DMA controller transmits the write data to the device in the form of one or more Data FIS. If more than one data FIS is required to complete the overall data transfer request, a DMA Activate FIS shall be sent prior to each and every one of the subsequent data FIS's. The amount of data transmitted to the device is determined by the transfer count programmed into the host adapter's DMA controller by the host driver software during the command setup phase.
- Upon completion of the transfer, the device transmits a Register – Device to Host FIS to indicate ending status for the command, clearing the BSY bit in the Status register, and if the interrupt flag is set in the header an interrupt is asserted to the host.

In some error conditions the device may signal an ending status by transmitting a register frame to the host without having transmitted a DMA Activate FIS. In such cases the host driver software should abort and clean up the DMA controller.

D.1.3 PIO data read from the device

- The host driver software issues a PIO read command to the device by writing the Shadow Register Block Registers (command register last).
- In response to the command register being written, the host adapter sets the BSY bit in the status shadow register and transmits a Register – Host to Device FIS to the device with the Shadow Register Block contents.
- When the device has processed the command and is ready to begin transferring data to the host, it first transmits a PIO Setup FIS to the host. Upon receiving the PIO Setup FIS, the host adapter holds the FIS contents in a temporary holding buffer.
- The device follows the PIO Setup FIS with a Data – Device to Host FIS. Upon receiving the Data FIS while holding the PIO Setup FIS, the host adapter transfers the register contents from the PIO Setup FIS into the shadow registers including the initial status value, resulting in DRQ getting set and BSY getting cleared in the Status register. Also, if the interrupt flag is set, an interrupt is generated to the host.
- The host controller receives the incoming data that is part of the Data FIS into a speed matching FIFO that is conceptually attached to the Shadow Register data Block Register.
- As a result of the issued interrupt and DRQ being set in the Status register, host software does a REP INSW on the data register and pulls data from the head of the speed matching FIFO while the serial link is adding data to the tail of the FIFO. The flow control scheme handles data throttling to avoid underflow/overflow of the receive speed matching FIFO that feeds the data Shadow Register Block Register.
- When the number of words read by host software from the Data shadow register reaches the value indicated in the PIO Setup FIS, the host transfers the ending status value from the earlier PIO Setup FIS into the status shadow register resulting in DRQ being cleared and the ending status reported.
- If there are more data blocks to be transferred, the ending status will indicate BSY, and the process will repeat from the device sending the PIO Setup FIS to the host.

D.1.4 PIO data write to the device

- The host driver software issues a PIO write command to the device by writing the Shadow Register Block Registers (command register last).
- In response to the command register being written, the host adapter sets the BSY bit in the status shadow register and transmits a Register – Host to Device FIS to the device with the Shadow Register Block contents.
- When the device is ready to receive the PIO write data, it transmits a PIO Setup FIS to the host to indicate that the target is ready to receive PIO data and the number of words of data that are to be transferred.
- In response to a PIO Setup FIS with the D bit indicating a write to the device, the host transfers the beginning Status register contents from the PIO Setup FIS into the shadow Status register, resulting in DRQ getting set, BSY cleared. Also, if the interrupt flag is set, an interrupt is generated to the host.
- As a result of DRQ being set in the shadow Status register, the host driver software starts a REP OUTSW to the data register.
- The data written to the data register is placed in an outbound speed matching FIFO and is transmitted to the device as a Data – Host to Device FIS. The REP OUTSW pushes data onto the tail of the FIFO and the serial link pulls data from the head. The flow control scheme handles data throttling to avoid underflow of the transmit FIFO.
- When the number of words indicated in the PIO Setup FIS have been written to the transmit FIFO, the host controller transfers the final status value indicated in the PIO Setup frame into the shadow Status register resulting in DRQ being cleared, and closes the frame with a CRC

and EOF. If additional sectors of data are to be transferred, the ending status value transferred to the status shadow register would have the BSY bit set and the state is the same as immediately after the command was first issued to the device.

- If there are more data blocks to be transferred, the ending status will indicate BSY, and the process will repeat from the device sending the PIO Setup FIS to the host.
- When the number of sectors indicated in the Sector Count register have been transferred, the device shall send a Register – Device to Host FIS with the command complete interrupt and not BSY status.
- In the case of a write error, the device may, on any sector boundary include error status and a command complete interrupt in the PIO setup FIS, and there is no need to send the Register – Device to Host FIS.

D.1.5 Queued DMA read from device

NOTE – Serial ATA devices may choose to not implement legacy ATA queuing in favor of a more efficient native Serial ATA queuing mechanism.

- Prior to the command being issued to the device, the host driver software programs the host-side DMA controller with the memory address pointer(s) and the transfer direction, and arms the DMA controller (enables the “run” flag).
- The host driver software issues the command to the device by writing the Shadow Register Block Registers (command register last).
- In response to the command Shadow Register Block Register being written, the host adapter sets the BSY bit in the status shadow register and transmits a Register – Host to Device FIS to the device with the Shadow Register Block contents.
- When the device has queued the command and wishes to release the bus, it transmits a register FIS to the host resulting in the BSY bit being cleared and the REL bit being set in the Status register.
- When the device is ready to complete the transfer for the queued command, it transmits a Set Device Bits FIS to the host resulting in the SERV bit being set in the Status register. If no other command is active (i.e. BSY set to one), then an interrupt is also generated.
- In response to the service request, the host software deactivates the DMA controller (if activated) and issues a SERVICE command to the device by writing the Shadow Register Block Registers, resulting in the BSY bit getting set and a register FIS being transmitted to the device.
- In response to the SERVICE request, the device transmits a register FIS to the host conveying the TAG value to the host and clearing the BSY bit and settings the DRQ bit.
- When the DRQ bit is set, the host software reads the TAG value from the Shadow Register Block and restores the DMA controller context appropriate for the command that is completing.
- The device transmits the read data to the host in the form of one or more Data FIS. This transfer proceeds in response to flow control signals/readiness. Any DMA data arriving before the DMA controller has its context restored will back up into the inbound speed-matching FIFO until the FIFO is filled and will thereafter be flow controlled to throttle the incoming data until the DMA controller has its context restored by host software.
- The host controller recognizes that the incoming packet is a Data FIS and the DMA controller is programmed, and directs the incoming data to the host controller’s DMA controller that forwards the incoming data to the appropriate host memory locations.
- Upon completion of the transfer, the target transmits a Register – Device to Host FIS to indicate ending status for the command, clearing the BSY bit in the Status register, and if the interrupt flag is set in the header an interrupt is asserted to the host.

D.1.6 Queued DMA write to device

NOTE – Serial ATA devices may choose to not implement legacy ATA queuing in favor of a more efficient native Serial ATA queuing mechanism.

- Prior to the command being issued to the device, the host driver software programs the host-side DMA controller with the memory address pointer(s) and the transfer direction, and arms the DMA controller (enables the “run” flag).
- The host driver software issues the command to the device by writing the Shadow Register Block Registers (command register last).
- In response to the Shadow Register command Block Register being written, the host adapter sets the BSY bit in the status shadow register and transmits a Register – Host to Device FIS to the device with the Shadow Register Block contents.

- When the device has queued the command and wishes to release the bus, it transmits a register FIS to the host resulting in the BSY bit being cleared and the REL bit being set in the Status register.
- When the device is ready to complete the transfer for the queued command, it transmits a Set Device Bits FIS to the host resulting in the SERV bit being set in the Status register. If no other command is active (i.e. BSY set to one), then an interrupt is also generated.
- In response to the service request, the host software deactivates the DMA controller (if activated) and issues a SERVICE command to the device by writing the Shadow Register Block Registers, resulting in the BSYT bit getting set and a register FIS being transmitted to the device.
- In response to the SERVICE request, the device transmits a register FIS to the host conveying the TAG value to the host and clearing the BSY bit and setting the DRQ bit.
- When the DRQ bit is set, the host software reads the TAG value from the Shadow Register Block and restores the DMA controller context appropriate for the command that is completing.
- When the device is ready to receive the data from the host, the device transmits a DMA Activate FIS to the host, which activates the armed DMA controller. The DMA controller transmits the write data to the device in the form of one or more Data – Host to Device FIS's. If more than one data FIS is required to complete the overall data transfer request, a DMA Activate FIS shall be sent prior to each and every one of the subsequent data FIS's. The amount of data transmitted to the device is determined by the transfer count programmed into the host's DMA controller by the host driver software during the command setup phase. If the DMA Activate FIS arrives at the host prior to host software restoring the DMA context, the DMA Activate FIS results in the DMA controller starting the transfer as soon as the host software completes programming it (i.e. the controller is already activated, and the transfer starts as soon as the context is restored).
- Upon completion of the transfer, the target transmits a Register – Host to Device FIS to indicate ending status for the command, clearing the BSY bit in the Status register, and if the interrupt flag is set in the header an interrupt is asserted to the host.

D.1.7 ATAPI Packet commands with PIO data in

- The host driver software issues a PACKET command to the device by writing the Shadow Register Block Registers (command register last).
- In response to the command register being written, the host adapter sets the BSY bit in the status shadow register and transmits a Register – Host to Device FIS to the device with the Shadow Register Block contents.
- When the device is ready to receive the ATAPI command packet, it transmits a PIO Setup – Device to Host FIS to the host to indicate that the target is ready to receive PIO data and the number of words of data that are to be transferred.
- The host transfers the beginning Status register contents from the PIO Setup FIS into the shadow Status register, resulting in BSY getting deasserted and DRQ getting asserted.
- As a result of BSY getting cleared and DRQ being set in the shadow Status register, the host driver software writes the command packet to the Shadow Register Block data register.
- The data written to the data register is placed in an outbound speed matching FIFO and is transmitted to the device as a Data – Host to Device FIS. The writes to the data register push data onto the tail of the FIFO and the serial link pulls data from the head. The flow control scheme handles data throttling to avoid underflow of the transmit FIFO.
- When the number of words indicated in the PIO Setup FIS have been written to the transmit FIFO, the host controller transfers the final status value indicated in the PIO setup frame into the shadow Status register resulting in DRQ being cleared and BSY being set, and closes the frame with a CRC and EOF. This completes the transmission of the command packet to the device.

- When the device has processed the command and is ready to begin transferring data to the host, it first transmits a PIO Setup – Device to Host FIS to the host. Upon receiving the PIO Setup – Device to Host FIS, the host adapter holds the FIS contents in a temporary holding buffer.
- The device follows the PIO Setup – Device to Host FIS with a Data – Device to Host FIS. Upon receiving the Data FIS while holding the PIO Setup FIS context, the host adapter transfers the register contents from the PIO Setup FIS into the shadow registers including the initial status value, resulting in DRQ getting set and BSY getting cleared in the Status register. Also, if the interrupt flag is set, an interrupt is generated to the host.
- The host controller receives the incoming data that is part of the Data FIS into a speed matching FIFO that is conceptually attached to the data Shadow Register Block Register.
- As a result of the issued interrupt and DRQ being set in the Status register, host software reads the byte count and does a REP INSW on the data register to pull data from the head of the speed matching FIFO while the serial link is adding data to the tail of the FIFO. The flow control scheme handles data throttling to avoid underflow/overflow of the receive speed matching FIFO that feeds the data Shadow Register Block Register.
- When the number of words received in the Data FIS reaches the value indicated in the PIO Setup FIS and the host FIFO is empty, the host transfers the ending status value from the earlier PIO Setup into the status shadow register resulting in DRQ being cleared and, BSY being set.
- The device transmits final ending status by sending a register FIS with BSY cleared to zero and the ending status for the command and the interrupt flag set.
- The host detects an incoming register frame that contains BSY cleared to zero and ending status for the command and the interrupt flag set and places the frame content into the shadow registers to complete the command.

D.1.8 ATAPI Packet commands with PIO data out

- The host driver software issues a PACKET command to the device by writing the Shadow Register Block Registers (command register last).
- In response to the command register being written, the host adapter sets the BSY bit in the status shadow register and transmits a Register – Host to Device FIS to the device with the Shadow Register Block contents.
- When the device is ready to receive the ATAPI command packet, it transmits a PIO Setup – Device to Host FIS to the host to indicate that the target is ready to receive PIO data and the number of words of data that are to be transferred.
- The host transfers the beginning Status register contents from the PIO Setup – Device to Host FIS into the shadow Status register, resulting in BSY getting deasserted and DRQ getting asserted.
- As a result of BSY getting cleared and DRQ being set in the shadow Status register, the host driver software writes the command packet to the Shadow Register Block data register.
- The data written to the data register is placed in an outbound speed matching FIFO and is transmitted to the device as a Data – Host to Device FIS. The writes to the data register push data onto the tail of the FIFO and the serial link pulls data from the head. The flow control scheme handles data throttling to avoid underflow of the transmit FIFO.
- When the number of words indicated in the PIO Setup FIS have been written to the transmit FIFO, the host controller transfers the final status value indicated in the PIO Setup frame into the shadow Status register resulting in DRQ being cleared and BSY being set, and closes the frame with a CRC and EOF. This completes the transmission of the command packet to the device.
- When the device has processed the command and is ready to receive the PIO write data, it transmits a PIO Setup – Device to Host FIS to the host to indicate that the target is ready to receive PIO data and the number of words of data that are to be transferred.

- In response to a PIO Setup FIS with the D bit indicating a write to the device, the host transfers the beginning Status register contents from the PIO Setup FIS into the shadow Status register, resulting in DRQ getting set. Also, if the interrupt flag is set, an interrupt is generated to the host.
- As a result of DRQ being set in the shadow Status register, the host driver software reads the byte count and starts a REP OUTSW to the data register.
- The data written to the data register is placed in an outbound speed matching FIFO and is transmitted to the device as a Data – host to device FIS. The REP OUTSW pushes data onto the tail of the FIFO and the serial link pulls data from the head. The flow control scheme handles data throttling to avoid underflow of the transmit FIFO.
- When the number of words indicated in the PIO Setup FIS have been written to the transmit FIFO, the host controller transfers the final status value indicated in the PIO Setup FIS into the shadow Status register resulting in DRQ being cleared, BSY being set, and closes the frame with a CRC and EOF.
- The device transmits final ending status by sending a Register – Device to Host FIS with BSY cleared to zero and the ending status for the command and the interrupt flag set.
- The host detects an incoming Register – Device to Host FIS that contains BSY cleared to zero and ending status for the command and the interrupt flag set and places the frame content into the shadow registers to complete the command.

D.1.9 ATAPI Packet commands with DMA data in

- Prior to the command being issued to the device, the host driver software programs the host-side DMA controller with the memory address pointer(s) and the transfer direction, and arms the DMA controller (enables the “run” flag).
- The host driver software issues a PACKET command to the device by writing the Shadow Register Block Registers (command register last).
- In response to the command register being written, the host adapter sets the BSY bit in the status shadow register and transmits a Register – Host to Device FIS to the device with the Shadow Register Block contents.
- When the device is ready to receive the ATAPI command packet, it transmits a PIO Setup – Device to Host FIS to the host to indicate that the target is ready to receive PIO data and the number of words of data that are to be transferred.
- The host transfers the beginning Status register contents from the PIO Setup FIS into the shadow Status register, resulting in BSY getting deasserted and DRQ getting asserted.
- As a result of BSY getting cleared and DRQ being set in the shadow Status register, the host driver software writes the command packet to the Shadow Register Block data register.
- The data written to the data register is placed in an outbound speed matching FIFO and is transmitted to the device as a Data – Host to Device FIS. The writes to the data register push data onto the tail of the FIFO and the serial link pulls data from the head. The flow control scheme handles data throttling to avoid underflow of the transmit FIFO.
- When the number of words indicated in the PIO Setup FIS have been written to the transmit FIFO, the host controller transfers the final status value indicated in the PIO setup frame into the shadow Status register resulting in DRQ being cleared and BSY being set, and closes the frame with a CRC and EOF. This completes the transmission of the command packet to the device.
- When the device has processed the command and is ready, it transmits the read data to the host in the form of a single Data FIS. This transfer proceeds in response to flow control signals/readiness.
- The host controller recognizes that the incoming packet is a Data FIS and the DMA controller is programmed, and directs the incoming data to the host controller’s DMA controller that forwards the incoming data to the appropriate host memory locations.

- Upon completion of the transfer, the target transmits a Register – Device to Host FIS to indicate ending status for the command, clearing the BSY bit in the Status register, and if the interrupt flag is set in the header an interrupt is asserted to the host.

D.1.10 ATAPI Packet commands with DMA data out

- Prior to the command being issued to the device, the host driver software programs the host-side DMA controller with the memory address pointer(s) and the transfer direction, and arms the DMA controller (enables the “run” flag). As a result the DMA controller becomes armed but remains paused pending a signal from the device to proceed with the data transfer.
- The host driver software issues a PACKET command to the device by writing the Shadow Register Block Registers (command register last).
- In response to the command register being written, the host adapter sets the BSY bit in the status shadow register and transmits a Register – Host to Device FIS to the device with the Shadow Register Block contents.
- When the device is ready to receive the ATAPI command packet, it transmits a PIO Setup – Device to Host FIS to the host to indicate that the target is ready to receive PIO data and the number of words of data that are to be transferred.
- The host transfers the beginning Status register contents from the PIO Setup FIS into the shadow Status register, resulting in BSY getting deasserted and DRQ getting asserted.
- As a result of BSY getting cleared and DRQ being set in the shadow Status register, the host driver software writes the command packet to the Shadow Register Block data register.
- The data written to the data register is placed in an outbound speed matching FIFO and is transmitted to the device as a Data – Host to Device FIS. The writes to the data register push data onto the tail of the FIFO and the serial link pulls data from the head. The flow control scheme handles data throttling to avoid underflow of the transmit FIFO.
- When the number of words indicated in the PIO Setup FIS have been written to the transmit FIFO, the host controller transfers the final status value indicated in the PIO Setup frame into the shadow Status register resulting in DRQ being cleared and BSY being set, and closes the frame with a CRC and EOF. This completes the transmission of the command packet to the device.
- When the device is ready to receive the data from the host, the device transmits a DMA Activate FIS to the host, which activates the armed DMA controller. The DMA controller transmits the write data to the device in the form of one or more Data FIS. The transfer proceeds in response to flow control signals/readiness. The amount of data transmitted to the device is determined by the transfer count programmed into the host’s DMA engine by the host driver software during the command setup phase.
- Upon completion of the transfer, the device transmits a Register – Device to Host FIS to indicate ending status for the command, clearing the BSY bit in the Status register, and if the interrupt flag is set in the header an interrupt is asserted to the host.

D.1.11 First-party DMA read of host memory by device

This section only outlines the basic First-party DMA read transaction and does not outline command sequences that utilize this capability.

- As a result of some condition and the target having adequate knowledge of host memory (presumably some prior state has been conveyed to the target), the target signals a DMA read request for a given address and given transfer count to the host by transmitting a First-Party DMA Setup FIS.
- Upon receiving the First-Party DMA Setup FIS, the host controller transfers the appropriate memory address, count value, and transfer direction into the host-side DMA controller and arms and activates the DMA controller (enables the “run” flag).

- In response to being set up, armed, and activated, the DMA controller retrieves data from host memory and transmits it to the device in the form of one or more Data – Host to Device FIS's. The transfer proceeds in response to flow control signals/readiness.

D.1.12 First-party DMA write of host memory by device

This section only outlines the basic first-party DMA write transaction and does not outline command sequences which utilize this capability.

- As a result of some condition and the target having adequate knowledge of host memory (presumably some prior state has been conveyed to the target), the target signals a DMA write request for a given address and given transfer count to the host by transmitting a First-party DMA Setup FIS.
- Upon receiving the First-party DMA Setup FIS, the host controller transfers the appropriate memory address, count value, and transfer direction into the host-side DMA controller and arms the DMA controller (enables the “run” flag).
- The device then transmits the data to the host in the form of one or more Data – Device to Host FIS’s. This DMA transfer proceeds in response to flow control signals/readiness.
- The host controller recognizes that the incoming packet is a Data FIS and directs the incoming data to the host controller’s DMA controller which forwards the incoming data to the appropriate host memory locations.

D.1.13 Odd word count considerations

This section outlines special considerations required to accommodate data transfers of an odd number of 16-bit word quantities. The considerations are separately outlined for each of the data transaction types. No accommodation in Serial ATA is made for the transfer of an odd number of 8-bit byte quantities.

D.1.13.1 Legacy DMA read from target for odd word count

- Prior to the command being issued to the device, the host driver software programs the host-side DMA controller with the memory address pointer(s) and the transfer direction, and arms the DMA controller (enables the “run” flag). The count for the DMA transfer is an odd number of word (16-bit) quantities.
- When the device has processed the corresponding command and is ready to transmit the data to the host, it does so in the form of one or more data FIS. Because the transfer count is odd, the last 32-bit Dword transmitted to the host has the high order 16 bits padded with zeroes. The CRC value transmitted at the end of the FIS is computed over the entire FIS including any pad bytes in the final transmitted symbol.
- The host controller receives the incoming data and the DMA controller directs the received data from the receive FIFO to the appropriate host memory locations. The DMA controller has a transfer granularity of a 16-bit word (consistent with the 8038i specification).
- Upon receiving the final 32-bit Dword of receive data, the DMA controller transfers the first half (low order 16 bits) to the corresponding final memory location at which point the DMA engine’s transfer count is exhausted. The DMA controller drops the high-order 16 bits of the final received Dword since it represents data received beyond the end of the requested DMA transfer. The dropped 16 high order bits corresponds with the 16 bits of transmission pad inserted by the sender.

D.1.13.2 Legacy DMA write by host to target for odd word count

- Prior to the command being issued to the device, the host driver software programs the host-side DMA controller with the memory address pointer(s) and the transfer direction, and arms the DMA controller (enables the “run” flag). The count for the DMA transfer is an odd number of word (16-bit) quantities.
- When the device has processed the corresponding command and is ready to receive the data from the host, it signals readiness with a DMA Activate FIS.
- Upon receiving the DMA Activate signal, the host transmits the data to the device in the form of one or more data FIS. Because the transfer count is odd, the DMA controller completes its data transfer from host memory to the transmit FIFO after filling only the low order 16-bits of the last Dword in the FIFO, leaving the upper 16 bits zeroed. This padded final Dword is transmitted as the final symbol in the data frame. The CRC value transmitted at the end of the FIS is computed over the entire FIS including any pad bytes in the final transmitted symbol.
- Having awareness of the command set and having decoded the current command, the device that receives the transmitted data has knowledge of the expected data transfer length. Upon receiving the data from the host, the device removes the 16-bit pad data in the upper 16 bits of the final 32-bit Dword of received data.

D.1.13.3 PIO data read from the device

- In response to decoding and processing a PIO read command with a transfer count for an odd number of 16-bit words, the device transmits the corresponding data to the host in the form of a single Data FIS. The device pads the upper 16-bits of the final 32-bit Dword of the last transmitted FIS in order to close the FIS. The CRC value transmitted at the end of the FIS is computed over the entire FIS including any pad bytes in the final transmitted symbol.
- Host driver software responsible for retrieving the PIO data is aware of the number of words of data it expects to retrieve from the taskfile data register and performs a REP INSW operation for an odd number of repetitions.
- Upon exhaustion of the REP INSW operation by the host driver software, the receive FIFO that interfaces with the data register has one 16-bit word of received data remaining in it that corresponds to the pad that the device included at the end of the transmitted frame. This remaining word of data left in the data register FIFO is flushed upon the next write of the command taskfile register or upon the receipt of the next data FIS from the device.

D.1.13.4 PIO data write to the device

- In response to decoding and processing a PIO write command with a transfer count for an odd number of 16-bit words, the device transmits a PIO Setup FIS to the host indicating it is ready to receive the PIO data and indicating the transfer count. The conveyed transfer count is for an odd number of 16-bit word quantities.
- Host driver software responsible for transmitting the PIO data is aware of the number of words of data it will write to the taskfile data register and performs a REP OUTSW operation for an odd number of repetitions.
- After the final write by the software driver to the taskfile data register, the transfer count indicated in the PIO Setup packet is exhausted, which signals the host controller to close the FIS. Since the transfer count was odd, the upper 16 bits of the final 32-bit Dword of data to transmit remains zeroed (pad) and the host controller closes the FIS after transmitting this final padded Dword. The CRC value transmitted at the end of the FIS is computed over the entire FIS including any pad bytes in the final transmitted symbol.
- Having awareness of the command set and having decoded the current command, the device that receives the transmitted data has knowledge of the expected data transfer length. Upon receiving the data from the host, the device removes the 16-bit pad data in the upper 16 bits of the final 32-bit Dword of received data.

D.1.13.5 First-party DMA read of host memory by device

- When the device wishes to initiate a first-party DMA read of host memory for an odd word count, it constructs a First-party DMA Setup FIS that includes the base address and the transfer count (for an odd number of words).
- Upon receiving the DMA Setup FIS, the host controller transfers the starting address and (odd) transfer count to the DMA controller and activates the DMA controller to initiate a transfer to the device of the requested data.
- The host transmits the data to the device in the form of one or more data FIS. Because the transfer count is odd, the DMA controller completes its data transfer from host memory to the transmit FIFO after filling only the low order 16-bits of the last Dword in the FIFO, leaving the upper 16 bits zeroed. This padded final Dword is transmitted as the final symbol in the data frame. The CRC value transmitted at the end of the FIS is computed over the entire FIS including any pad bytes in the final transmitted symbol.
- Having awareness of the amount of data requested, the device that receives the transmitted data has knowledge of the expected data transfer length. Upon receiving the data from the host, the device removes the 16-bit pad data in the upper 16 bits of the final 32-bit Dword of received data.

D.1.13.6 First-party DMA write of host memory by device

- When the device wishes to initiate a first-party DMA write to host memory for an odd word count, it constructs a DMA Setup FIS that includes the base address and transfer count (for an odd number of words).
- Upon receiving the DMA Setup FIS, the host controller transfers the starting address and (odd) transfer count to the DMA controller and activates the DMA controller.
- The device transmits the data to the host in the form of one or more data FIS. Because the transfer count is odd, the last 32-bit Dword transmitted to the host has the high order 16 bits padded with zeroes. The CRC value transmitted at the end of the FIS is computed over the entire FIS including any pad bytes in the final transmitted symbol.
- The host controller receives the incoming data and the DMA controller directs the received data from the receive FIFO to the appropriate host memory locations. The DMA controller has a transfer granularity of a 16-bit word (consistent with the 8038i specification).
- Upon receiving the final 32-bit Dword of receive data, the DMA controller transfers the first half (low order 16 bits) to the corresponding final memory location at which point the DMA controller's transfer count is exhausted. The DMA controller drops the high-order 16 bits of the final received Dword since it represents data received beyond the end of the requested DMA transfer. The dropped 16 high order bits corresponds with the 16 bits of transmission pad inserted by the sender.

File 002 - https://web.archive.org/web/20061025163622/http:/www.eetchina.com/ARTICLES/2006AUG/PDF/HDMI_Spec_1.3_GM1.pdf?SOURCES=DOWNLOAD

High-Definition Multimedia Interface

Specification Version 1.3

Hitachi, Ltd.

Matsushita Electric Industrial Co., Ltd.

Philips Consumer Electronics, International B.V.

Silicon Image, Inc.

Sony Corporation

Thomson Inc.

Toshiba Corporation

June 22, 2006

Preface

Notice

THIS SPECIFICATION IS PROVIDED “AS IS” WITH NO WARRANTIES WHATSOEVER, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, NO WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION, OR SAMPLE.

Hitachi, Ltd., Matsushita Electric Industrial Co., Ltd., Philips Consumer Electronics International B.V., Silicon Image, Inc., Sony Corporation, Thomson Inc., Toshiba Corporation and HDMI Licensing, LLC disclaim all liability, including liability for infringement of any proprietary rights, relating to use of information in this specification.

Copyright © 2001-2006 by Hitachi, Ltd., Matsushita Electric Industrial Co., Ltd., Philips Consumer Electronics International, B.V., Silicon Image, Inc., Sony Corporation, Thomson Inc., and Toshiba Corporation. All rights reserved. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted herein. Unauthorized use or duplication prohibited. “HDMI” and all associated logos are trademarks of HDMI Licensing, LLC. Third-party trademarks and servicemarks are property of their respective owners.

Document Revision History

1.3	2006/06/22	Significant new features: - Type C “Mini-Connector” (4.1.9.5, 4.1.9.6) - Cable Categories 1 and 2 (4.2.6) - Deep Color [4:4:4] (6.5, 8.3.2) - Reference Cable Equalizer (4.2.3.2, 4.2.5, 4.2.6) - Higher-speed single-link (4.1.2, 4.2.3,through 4.2.6, 8.3.2) - xvYCC Enhanced Colorimetry (6.7.2.3) - Gamut Metadata transmission (5.3.12, 6.7.3, Appendix E)) - DST audio format (5.3.10, 7.6.3) - High-bitrate compressed audio formats (5.3.11, 7.2.4, 7.3.3, 7.6.2) - Auto-Lipsync Correction feature (8.3.2, 8.9) Updated normative reference from CEA-861-B to –D (1.2, throughout). Updated Overview for new features (3) Several minor editorial (throughout)
1.2a	2005/12/14	Changes to CEC supplement (see supplement for details) Eliminated I_{OFF} and made V_{OFF} normative (4.2.4) Changed CEC resistance to 5 ohms (4.2.10) Clarified DVI device discrimination (8.3.3) Several minor editorial (throughout)
1.2	2005/08/22	Removed limitations on Type A connector usage (4.1.2, 6.1) Required new connector mechanical features, optional in 1.1 (4.1.9) Required Sink support for future AC-coupled Sources (4.2.5) Add note regarding maximum ratings of Sink (4.2.5) Clarified Cable Assembly use of +5V Power (4.2.7) Removed incorrect testing method for DDC capacitance (4.2.8) Clarified when separate CEC lines on inputs are allowed (4.2.10) Add maximum resistance spec for interconnected CEC line (4.2.10) Remove CEC leakage current limit while in standby (4.2.10) Relaxed YC_BC_R output requirement for RGB devices (6.2.3) Added support for additional video formats (6.2.4, and 7.3.3, 8.2.1) Corrected sample rate requirement from 1000 ppm to $\pm$1000 ppm (7.2.6) Clarified use of Speaker Allocation Data Block (7.4) Added support for One Bit audio (7.9, and throughout) Clarified exception for 640x480p (VGA) declaration in EDID (8.3.4) Loosened requirement for duplicated DTD declarations (8.3.4) Added recommendation for setting Supports_AI (9.2) Clarified the behavior of Repeater to Sink with Supports_AI (9.3.2) Clarified rule for DVD-Audio ACP Packet transmission (9.3.5) Additional minor editorial (throughout)
1.1	2004/05/20	Permitted multi-rate native format support on Type A Sinks (4.1.2) Changed connector mechanical spec (4.1.9) Changed connector electrical spec (4.1.7) Removed CEC / +5V Power dependency for Source (4.2.7) Loosened regulation requirements for +5V Power (4.2.7)

DELL/HP/LENOVO EXHIBIT 1015
Page 359

		Made HPD voltages consistent with new +5V Power (4.2.9) Clarified CEC connection requirements (4.2.10) Restricted CTLx values allowed in non-Preamble periods (5.2.1) Added new Packet Types (5.3.1) Clarified InfoFrame Packet requirements (5.3.5) Added ACP and ISRC Packet definitions and usage (5.3.7, 8.8, 9.3) Specified recommended handling of non-Subpacket 0 CS blocks (7.1) Clarified audio sample rate requirements (7.2.6) Disallowed Layout 1 2-channel (7.6) Clarified AVI transmission requirements (8.2.1) Added extension fields and clarified HDMI VSDB (8.3.2) Clarified DVI/HDMI device discrimination (8.3.3) Clarified HPD behavior (8.5) Clarified EDID values of Physical Addresses (8.7) Made minor editorial changes (throughout)
1.0	2002/12/09	Initial Release

DELL/HP/LENOVO EXHIBIT 1015
Page 360

Intellectual Property Statement

Hitachi, Ltd., Matsushita Electric Industrial Co., Ltd., Philips Consumer Electronics International, B.V., Silicon Image, Inc., Sony Corporation, Thomson Inc., and Toshiba Corporation each may have patents and/or patent applications related to the High-Definition Multimedia Interface Specification. These companies have made available to the industry an Adopter Agreement that includes a limited, reciprocal patent license to certain of the electrical interfaces, mechanical interfaces, signals, signaling and coding protocols, and bus protocols described in the mandatory portions of the High-Definition Multimedia Interface Specification Release 1.3 published by HDMI Licensing, LLC.

Contact Information

The URL for the HDMI Founders web site is: http://www.HDMI.org.

Contribution

Silicon Image, Inc has made a significant contribution to this standard by editing the specification and developing the core technologies upon which this specification is based; including Transition Minimized Differential Signaling (TMDS®) technology.

Acknowledgement

HDMI founders acknowledge the concerted efforts of employees of Japan Aviation Electronics Industry, Limited and Molex Japan, who have made a significant contribution to this standard by developing the connector technology and the mechanical and electrical specifications for the required plugs and receptacles.

Table of Contents

PREFACE ..... **II**

NOTICE II

DOCUMENT REVISION HISTORY III

INTELLECTUAL PROPERTY STATEMENT V

CONTACT INFORMATION V

CONTRIBUTION V

ACKNOWLEDGEMENT V

1 INTRODUCTION **1**

1.1 PURPOSE AND SCOPE 1

1.2 NORMATIVE REFERENCES 1

1.3 INFORMATIVE REFERENCES 2

1.4 ORGANIZATION OF THIS DOCUMENT 2

1.5 USAGES AND CONVENTIONS 3

2 DEFINITIONS **4**

2.1 CONFORMANCE LEVELS 4

2.2 GLOSSARY OF TERMS 4

2.3 ACRONYMS AND ABBREVIATIONS 6

3 OVERVIEW **8**

4 PHYSICAL LAYER **10**

4.1 CONNECTORS AND CABLES 10

4.1.1 Overview of Connectors *10*

4.1.2 Connector Support Requirements *10*

4.1.3 Dual-Link *10*

4.1.4 Connector Pin Assignments *11*

4.1.5 Contact sequence *14*

4.1.6 Connector Mechanical Performance *15*

4.1.7 Connector Electrical Characteristics *17*

4.1.8 Connector Environmental Characteristics 19
4.1.9 Connector Drawings – Mating Interface Dimensions 21
4.1.10 Cable Adapter Specification 32
4.2 ELECTRICAL SPECIFICATION 38
4.2.1 TMDS Overview 38
4.2.2 TMDS System Operating Conditions 39
4.2.3 TMDS Specification and Testing Overview 40
4.2.4 HDMI Source TMDS Characteristics 42
4.2.5 HDMI Sink TMDS Characteristics 45
4.2.6 Cable Assembly TMDS Characteristics 47
4.2.7 +5V Power Signal 51
4.2.8 DDC 52
4.2.9 Hot Plug Detect Signal (HPD) 52
4.2.10 CEC Line 53
4.2.11 Robustness Requirements 54
5 SIGNALING AND ENCODING 55
5.1 OVERVIEW 55
5.1.1 Link Architecture 55
5.1.2 Operating Modes Overview 56
5.2 OPERATING MODES 57
5.2.1 Control Period 57
5.2.2 Video Data Period 59
5.2.3 Data Island Period 59
5.3 DATA ISLAND PACKET DEFINITIONS 63
5.3.1 Packet Header 63
5.3.2 Null Packet 64
5.3.3 Audio Clock Regeneration Packet 65
5.3.4 Audio Sample Packet 65
5.3.5 InfoFrame Packet 67
5.3.6 General Control Packet 68
5.3.7 Audio Content Protection Packet (ACP) 69
5.3.8 ISRC Packets 69
5.3.9 One Bit Audio Sample Packet 73

5.3.10 DST Audio Packet *74*
5.3.11 High-Bitrate (HBR) Audio Stream Packet *75*
5.3.12 Gamut Metadata Packet *75*
5.4 ENCODING 79
5.4.1 Serialization *79*
5.4.2 Control Period Coding *80*
5.4.3 TERC4 Coding *80*
5.4.4 Video Data Coding *81*
6 VIDEO 85
6.1 OVERVIEW 85
6.2 VIDEO FORMAT SUPPORT 85
6.2.1 Format Support Requirements *85*
6.2.2 Video Control Signals : HSYNC, VSYNC *86*
6.2.3 Pixel Encoding Requirements *86*
6.2.4 Color Depth Requirements *86*
6.3 VIDEO FORMAT TIMING SPECIFICATIONS 87
6.3.1 Primary Video Format Timings *87*
6.3.2 Secondary Video Format Timings *87*
6.4 PIXEL-REPETITION 88
6.5 PIXEL ENCODINGS AND COLOR DEPTH 89
6.5.1 24-Bit Pixel Encodings *89*
6.5.2 Deep Color Pixel Packing *90*
6.5.3 Deep Color Mode / Phase Indication *93*
6.5.4 Pixel Repetition *95*
6.6 VIDEO QUANTIZATION RANGES 96
6.7 COLORIMETRY 97
6.7.1 Default Colorimetry *97*
6.7.2 Applicable Colorimetry Standards *97*
6.7.3 Gamut-Related Metadata *98*
7 AUDIO 99
7.1 RELATIONSHIP WITH IEC 60958/IEC 61937 99
7.2 AUDIO SAMPLE CLOCK CAPTURE AND REGENERATION 99

7.2.1 N parameter *101*
7.2.2 CTS parameter *101*
7.2.3 Recommended N and Expected CTS Values *101*
7.2.4 L-PCM and IEC 61937 Compressed Audio ACR *103*
7.2.5 One Bit Audio ACR *103*
7.2.6 DST Audio ACR *103*
7.3 AUDIO SAMPLE RATES AND SUPPORT REQUIREMENTS 103
7.3.1 One Bit Audio Sample Rate Requirements *105*
7.3.2 DST Audio Sample Rate Requirements *105*
7.3.3 Video Dependency *106*
7.4 CHANNEL / SPEAKER ASSIGNMENT 107
7.5 AUDIO, VIDEO SYNCHRONIZATION 107
7.6 AUDIO DATA PACKETIZATION 107
7.6.1 One Bit Audio Packetization *109*
7.6.2 High-Bitrate Audio Stream Packetization *110*
7.6.3 DST Packetization *110*
7.7 ERROR HANDLING (INFORMATIVE) 110
7.8 PACKET DELIVERY RULES 110
7.8.1 Audio Sample Packets *110*
7.8.2 Audio Clock Regeneration Packets *111*
7.9 ONE BIT AUDIO USAGE OVERVIEW 111
7.10 DST USAGE OVERVIEW 111

8 CONTROL AND CONFIGURATION 112

8.1 OVERVIEW 112
8.2 INFOFRAMES 112
8.2.1 Auxiliary Video information (AVI) InfoFrame *112*
8.2.2 Audio InfoFrame *115*
8.3 E-EDID DATA STRUCTURE 118
8.3.1 CEA Extension *118*
8.3.2 HDMI Vendor-Specific Data Block (HDMI VSDB) *118*
8.3.3 DVI/HDMI Device Discrimination *121*
8.3.4 Audio and Video Details *121*
8.4 ENHANCED DDC 121

8.4.1 Timing 122
8.4.2 Data Transfer Protocols 122
8.4.3 Segment pointer 122
8.4.4 Enhanced DDC Sink 122
8.4.5 Enhanced DDC Source 122
8.5 HOT PLUG DETECT SIGNAL 123
8.6 CONSUMER ELECTRONICS CONTROL (CEC) 123
8.7 PHYSICAL ADDRESS 123
8.7.1 Overview 123
8.7.2 Physical Address Discovery 124
8.7.3 Discovery Algorithm 126
8.7.4 HDMI Sink Query 126
8.8 ISRC HANDLING 126
8.9 AUTO LIPSYNC CORRECTION FEATURE 127
8.9.1 EDID Latency Info 128
8.9.2 Compensation 128
8.9.3 Supporting Dynamic Latency Changes 129
8.9.4 Separate Audio and Video Paths 129

9 CONTENT PROTECTION 130

9.1 RECOMMENDATION 130
9.2 HDCP IMPLEMENTATIONS 130
9.3 USAGE OF AUDIO CONTENT PROTECTION (ACP) PACKETS 130
9.3.1 Requirements for Sink 130
9.3.2 Requirements for Repeater 131
9.3.3 Application to Generic Audio 131
9.3.4 Application to IEC 60958-Identified Audio 131
9.3.5 Application to DVD-Audio 131
9.3.6 Application to Super Audio CD 132

APPENDIX A REPEATER 134

A.1 REPEATER FUNCTIONS 134
A.2 E-EDID READ TIMING (INFORMATIVE) 134

APPENDIX B TYPE B CONNECTOR USAGE 135

B.1 EXCEPTION TO AUDIO FORMAT SUPPORT REQUIREMENT 135
B.2 HDMI DUAL-LINK ARCHITECTURE 135

APPENDIX C COMPATIBILITY WITH DVI 136

C.1 REQUIREMENT FOR DVI COMPATIBILITY 136
C.2 HDMI SOURCE REQUIREMENTS 136
C.3 HDMI SINK REQUIREMENTS 137
C.4 TYPE A TO DVI ADAPTER CABLE [INFORMATIVE] 137
C.5 TYPE B TO DVI ADAPTER CABLE [INFORMATIVE] 139

APPENDIX D DEEP COLOR ADDITIONAL DETAIL 140

E.1 STATE MACHINES 140
D.2 RECOMMENDED N AND EXPECTED CTS VALUES 141

APPENDIX E GAMUT-RELATED METADATA 146

E.1 OVERVIEW 146
E.2 TRANSMISSION PROFILES 146
E.3 GAMUT BOUNDARY DESCRIPTION 146
E.4 DATA PACKING 149
E.5 EXAMPLE P0 DATA STRUCTURES 150

APPENDIX F VIDEO SCALING AUTO-CONFIGURATION 153

SUPPLEMENT 1 CONSUMER ELECTRONICS CONTROL (CEC) ..... CEC-I

SEE SUPPLEMENT FOR TABLE OF CONTENTS

Figures

Figure 3-1 HDMI Block Diagram 8
Figure 4-1 Type A Receptacle Mating Interface Dimensions 21
Figure 4-2 Type A Plug Mating Interface Dimensions 23
Figure 4-3 Type A Receptacle and Plug Mated Condition 24
Figure 4-4 Type A Cable – Minimum Recommended Cable Bend 24
Figure 4-5 Type B Receptacle Mating Interface Dimensions 25
Figure 4-6 Type B Plug Mating Interface Dimensions 28
Figure 4-7 Type B Receptacle and Plug Mated Condition 28
Figure 4-8 Type C Receptacle Mating Interface Dimensions 29
Figure 4-9 Type C Plug Mating Interface Dimensions 30
Figure 4-10 Type C Receptacle and Plug Mated Condition 31
Figure 4-11 Type C Product Design Recommendations 31
Figure 4-12 Conceptual Schematic for one TMDS differential pair 38
Figure 4-13 Single-ended Differential Signal 39
Figure 4-14 Differential Signal 39
Figure 4-15 TMDS Link Test Points 39
Figure 4-16 Gain of Reference Cable Equalizer 42
Figure 4-17 Balanced Source Test Load 42
Figure 4-18 Eye Diagram Mask at TP1 for Source Requirements 44
Figure 4-19 HDMI Sink Test Points 45
Figure 4-20 Eye Diagram Mask at TP2 for Sink Requirements 47
Figure 4-21 Cable Assembly Test Points 48
Figure 4-22 Category 1 Cable Attenuation Limits – Sufficient Condition 49
Figure 4-23 Category 2 Cable Attenuation Limits – Sufficient Condition 50
Figure 4-24 Category 2 Additional Attenuation Limits For Copper Cable 50
Figure 4-25 Category 2 Additional Attenuation Limits for Equalized Cable 51
Figure 5-1 HDMI Encoder/Decoder Overview 55
Figure 5-2 Informative Example: TMDS periods in 720x480p video frame 56
Figure 5-3 TMDS Periods and Encoding 60
Figure 5-4 Data Island Packet and ECC Structure 62
Figure 5-5 Error Correction Code generator 63

Figure 5-6 Example P0 Transmission Sequence 77
Figure 5-7 TMDS Video Data Encode Algorithm 83
Figure 5-8 TMDS Video Decode Algorithm 84
Figure 6-1 Default pixel encoding: RGB 4:4:4, 8 bits/component 89
Figure 6-2 YC_BC_R 4:2:2 component 90
Figure 6-3 8-bit YC_BC_R 4:4:4 mapping 90
Figure 6-4 YC_BC_R 4:2:2 with Pixel-Doubling 96
Figure 7-1 Audio Clock Regeneration model 100
Figure 7-2 Optional Implementation: Audio Sink 100
Figure 7-3 Example Audio Sample Timing (Informative) 109
Figure 8-1 CEC and DDC line connections 123
Figure 8-2 Typical HDMI cluster 124
Figure 8-3 Addresses within an HDMI cluster 125
Figure 8-4 ISRC/CCI and ISRC Status Handling 127

Tables

Table 4-1 Type A Connector Pin Assignment 11
Table 4-2 Type B Connector Pin Assignment 12
Table 4-3 Type C Connector Pin Assignment 13
Table 4-4 Connector Contact Sequence 14
Table 4-5 Type A and Type C Plug and Receptacle Mechanical Performance 15
Table 4-6 Electrical Performance 17
Table 4-7 Connector Environmental Performance 19
Table 4-8 Wire Categories 32
Table 4-9 Type A-to-Type A Cable Wire Assignment 33
Table 4-10 Type A-to-Type B Cable Wire Assignment 34
Table 4-11 Type B to Type B Cable Wire Assignment 35
Table 4-12 Type C-to-Type C Cable Wire Assignment 36
Table 4-13 Type C-to-Type A Cable Wire Assignment 37
Table 4-14 Required Operating Conditions for HDMI Interface (see Figure 4-12) 40
Table 4-15 Source DC Characteristics at TP1 43
Table 4-16 Source AC Characteristics at TP1 43
Table 4-17 Sink Operating DC Characteristics at TP2 45
Table 4-18 Sink DC Characteristics When Source Disabled or Disconnected at TP2 46
Table 4-19 Sink AC Input Characteristics at TP2 46
Table 4-20 HDMI Sink Impedance Characteristics at TP2 47
Table 4-21 Cable Assembly TMDS Parameters 49
Table 4-22 +5V Power Pin Voltage 52
Table 4-23 Maximum Capacitance of DDC line 52
Table 4-24 Pull-up Resistance on DDC Lines 52
Table 4-25 Required Output Characteristics of Hot Plug Detect Signal 53
Table 4-26 Required Detection Levels for Hot Plug Detect Signal 53
Table 4-27 CEC line Electrical Specifications for all Configurations 54
Table 5-1 Encoding Type and Data Transmitted 57
Table 5-2 Preambles for Each Data Period Type 58
Table 5-3 TMDS Link Timing Parameters 59
Table 5-4 Extended Control Period Parameters 59

Table 5-5 Video Leading Guard Band Values 59
Table 5-6 Data Island Leading and Trailing Guard Band Values 61
Table 5-7 Packet Header 63
Table 5-8 Packet Types 64
Table 5-9 Null Packet Header 64
Table 5-10 Audio Clock Regeneration Packet Header 65
Table 5-11 Audio Clock Regeneration Subpacket 65
Table 5-12 Audio Sample Packet Header 66
Table 5-13 Audio Sample Subpacket 66
Table 5-14 InfoFrame Packet Header 67
Table 5-15 InfoFrame Packet Contents 67
Table 5-16 General Control Packet Header 68
Table 5-17 General Control Subpacket 68
Table 5-18 ACP Packet Header 69
Table 5-19 ACP Packet contents 69
Table 5-20 ISRC1 Packet Header 70
Table 5-21 ISRC1 Packet contents 71
Table 5-22 ISRC2 Packet Header 72
Table 5-23 ISRC2 Packet contents 72
Table 5-24 One Bit Audio Packet Header 73
Table 5-25 One Bit Audio Subpacket 74
Table 5-26 DST Audio Packet Header 74
Table 5-27 DST Audio Packet Body 75
Table 5-28 HBR Audio Stream Packet Header 75
Table 5-29 Gamut Boundary Description Metadata Profiles 76
Table 5-30 Gamut Metadata Packet Header 78
Table 5-31 Gamut Metadata Packet for P0 Transmission Profile 79
Table 5-32 Gamut Metadata Packet for P1 and Higher – 1st Packet of Sequence 79
Table 5-33 Gamut Metadata Packet for P1 and Higher – Remaining Packets 79
Table 5-34 Control-signal Assignment 80
Table 5-35 Encoding Algorithm Definitions 82
Table 6-1 Color Depth (CD field) Values 94
Table 6-2 Pixel Packing Phase (PP field) Values 95

Table 6-3 Video Color Component Ranges 96
Table 7-1 Recommended N and Expected CTS for 32kHz 102
Table 7-2 Recommended N and Expected CTS for 44.1kHz and Multiples 102
Table 7-3 Recommended N and Expected CTS for 48kHz and Multiples 103
Table 7-4 Allowed Values for Channel Status bits 24 to 27 105
Table 7-5 Max Sampling Frequency for 24-bit Video Format Timings* (Informative) 106
Table 7-6 Audio Packet Layout and Layout Value 108
Table 7-7 Valid Sample_Present Bit Configurations for Layout 0 108
Table 8-1 AVI InfoFrame Packet Header 113
Table 8-2 AVI InfoFrame Packet Contents 113
Table 8-3 HDMI Valid Pixel Repeat Values for Each Video Format Timing 115
Table 8-4 Audio InfoFrame Packet Header 116
Table 8-5 Audio InfoFrame Packet contents 117
Table 8-6 HDMI-LLC Vendor-Specific Data Block (HDMI VSDB) 119
Table 9-1 ACP_Type Dependent Fields for DVD-Audio Application 132
Table 9-2 ACP_Type Dependent Fields for Super Audio CD Application 133

1 Introduction

1.1 Purpose and Scope

This document constitutes the specification for the High-Definition Multimedia Interface (HDMI), version 1.3.

The High-Definition Multimedia Interface is provided for transmitting digital television audiovisual signals from DVD players, set-top boxes and other audiovisual sources to television sets, projectors and other video displays.

HDMI can carry high quality multi-channel audio data and can carry all standard and high-definition consumer electronics video formats. Content protection technology is available.

HDMI can also carry control and status information in both directions.

This specification completely describes the interface such that one could implement a complete transmission and interconnect solution or any portion of the interface. The underlying Transition Minimized Differential Signaling (TMDS)-based protocol and associated electrical signaling is described in detail. The mechanical specification of the connector and the signal placement within the connector are described.

A device that is compliant with this specification is interoperable with other compliant devices through the configuration and implementation provided for in this specification.

Mechanical, electrical, behavioral and protocol requirements necessary for compliance are described for sources, sinks and cables.

1.2 Normative References

The following standards contain provisions that, through reference in this text, constitute normative 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. If the referenced standard is dated, the reader is advised to use the version specified.

CEA, CEA-861-D, “A DTV Profile For Uncompressed High Speed Digital Interfaces”[1]

VESA, VESA E-EDID Standard, ENHANCED EXTENDED DISPLAY IDENTIFICATION DATA STANDARD Release A, Revision 1, February 9, 2000

VESA, VESA E-DDC Standard, ENHANCED DISPLAY DATA CHANNEL STANDARD Version 1, September 2, 1999

Philips Semiconductors, The I^2C-bus Specification, Version 2.1, January 2000

[1] All HDMI devices are required to comply with the requirements specified in CEA-861-D except where specifically noted in this document. The CEA-861-D term “source” should be read as “(HDMI) Source” and the terms “Display”, “Monitor” or “DTV Monitor” should be read as “(HDMI) Sink”.

ITU, ITU-R BT.601-5 Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios (October 1995)

ITU, ITU-R BT.709-5 Parameter values for the HDTV standards for production and international programme exchange (2002)

IEC, IEC 60958-1, “Digital audio interface – Part 1: General”, First edition 1999-12

IEC, IEC 60958-3, “Digital audio interface – Part 3: Consumer applications”, Third edition 2006-05[pw9]

IEC, IEC 61937, “Digital Audio - Interface for non-linear PCM encoded audio bitstreams applying IEC 60958”, First edition 2000-04

IEC, IEC 61966-2-4: “Multimedia systems and equipment - Colour measurement and management - Part 2-4: Colour management - Extended-gamut YCC colour space for video applications – xvYCC”, January 2006

DDWG, “Digital Visual Interface,” Revision 1.0, April 2, 1999 (DVI)

DVD Forum, “DVD Specifications for Read-Only Disc”, “Part 4: AUDIO SPECIFICATIONS”, Version 1, March 1999.

DVD Forum, “DVD Specifications for Read-Only Disc”, “Part 4: AUDIO SPECIFICATIONS”, Version-up Information (from 1.1 to 1.2), May 2000.

Digital Content Protection LLC, “High-bandwidth Digital Content Protection System Specification”, Revision 1.10 (HDCP)

Royal Philips Electronics and SONY Corporation, “Super Audio CD System Description Version 2.0”

1.3 Informative References

The following documents contain information that is useful in understanding this standard. Some of these documents are drafts of standards that may become normative references in a future release of this standard.

ANSI/SMPTE, SMPTE Standard 170M (1999) for Television – Composite Analog Video Signal – NTSC for Studio Applications

ANSI/SMPTE, SMPTE Standard 274M

ANSI/SMPTE, SMPTE Standard 296M

1.4 Organization of this document

This specification is organized as follows:

- Chapter 1 introduces HDMI, describes the purpose and scope of this document, references, organization of the document and usages and conventions.
- Chapter 2 defines terms and acronyms used throughout this document.
- Chapter 3 provides a high-level overview of the operation of HDMI.

- Chapter 4 describes the details of the Physical Layer of HDMI including basic electrical specifications and mechanical specifications of cables and connectors.
- Chapter 5 describes the Signaling and Encoding used by HDMI including descriptions of the different periods and encoding types used to transmit audio, video, and control data types and packet definitions for audio and auxiliary data.
- Chapter 6 describes Video related issues including video format timings, pixel encodings (RGB, YC_BC_R), colorimetry and corresponding requirements.
- Chapter 7 describes Audio related issues including audio clock regeneration, placement of audio samples within packets, packet timing requirements, audio sample rates and requirements, and channel/speaker assignments.
- Chapter 8 describes Control and Configuration functions, mechanisms and requirements, including use of the E-EDID, and InfoFrames.
- Chapter 9 describes the Content protection used for HDMI.
- Appendix A describes the usage of Repeaters and Switches.
- Appendix B describes restrictions related to the use of the Type B connector.
- Appendix C describes DVI compatibility.
- Appendix D describes additional details of Deep Color not covered in chapter 6, including example state machines and details of audio transmission in a Deep Color mode.
- Appendix E describes data structures and characteristics of gamut boundary descriptions, used when transmitting video using xvYCC colorimetry.
- Appendix F recommends a number of rules regarding a Sink's indication of video format support and automatic configuration of video output format for a Source.
- Supplement 1 describes use of the Consumer Electronics Control (CEC) line and protocol.

1.5 Usages and Conventions

bit N	Bits are numbered in little-endian format, i.e. the least-significant bit of a byte or word is referred to as bit 0.
D[X:Y]	Bit field representation covering bit X to bit Y (inclusive) of value or field D.
0xNN	Hexadecimal representation of base-16 numbers are represented using 'C' language notation, preceded by '0x'.
0bNN	Binary (base-2) numbers are represented using 'C' language notation, preceded by '0b'.
NN	Decimal (base-10) numbers are represented using no additional prefixes or suffixes.

Within this specification, any descriptions of data structures, values or sequences that occur on the HDMI interface should be interpreted only as data structures, values and sequences that are transmitted by the HDMI Source. Due to the possibility of errors during the transmission, these items should not be construed as data structures, values or sequences that are guaranteed to be detected by the HDMI Sink.

2 Definitions

2.1 Conformance Levels

expected	A key word used to describe the behavior of the hardware or software in the design models *assumed* by this specification. Other hardware and software design models may also be implemented.
may	A key word that indicates flexibility of choice with *no implied preferenc*e.
shall	A key word indicating a mandatory requirement. Designers are *required* to implement all such mandatory requirements.
should	A key word indicating flexibility of choice with a strongly preferred alternative. Equivalent to the phrase *is recommended*.
reserved fields	A set of bits within a data structure that are defined in this specification as reserved, and are not otherwise used. Implementations of this specification shall zero these fields. Future revisions of this specification, however, may define their usage.
reserved values	A set of values for a field that are defined in this specification as reserved, and are not otherwise used. Implementations of this specification shall not generate these values for the field. Future revisions of this specification, however, may define their usage.

2.2 Glossary of Terms

(Audio) Channel	Audio data intended to be delivered to a single audio speaker.
(Audio) Sample Clock	Original clock related to the audio input samples at the Source or the generated clock used to time the output of audio samples.
BCH	Error correction technique named after the developers: Bose, Chauduri, and Hocquenghem.
Byte	Eight bits of data.
CEA Extension	A 128 byte EDID 1.3-compatible extension block defined in CEA-861-D, designed to allow declaration of audio formats, additional video formats (beyond those in the base EDID structure) and other characteristics of the Sink..
CEC Root (Device)	A device, generally a display (Sink) device, formally defined by the following rule: A device that has no HDMI output or, a device that has chosen to take the physical address 0.0.0.0 (see Section 8.7).
Compressed (Audio)	All audio formats carried by HDMI other than L-PCM and One Bit Audio.
Data Stream Disparity	Integer indicating “DC-offset” level of link. A positive value represents the excess number of “1”s that have been transmitted. A negative value represents the excess number of “0”s that have been transmitted.

Direct Stream Transport	An audio format which is a lossless compression of Direct Stream Digital (DSD), as used in SuperAudio CD. DST is described in ISO/IEC 14496, part 3, Amendment 6: Lossless coding of oversampled audio.
Downstream	In the direction of the primary audio and video data flow, i.e. towards the Sink (e.g. display).
DVD-Audio	Disk format conforming to any version of “DVD Specifications for Read-Only Disc”, “Part 4: AUDIO SPECIFICATIONS”.
(HDMI) Source	A device with an HDMI output.
(HDMI) Sink	A device with an HDMI input.
(HDMI) Repeater	A device with one or more HDMI inputs and one or more HDMI outputs. Repeater devices shall simultaneously behave as both an HDMI Sink and an HDMI Source.
InfoFrame	A data structure defined in CEA-861-D that is designed to carry a variety of auxiliary data items regarding the audio or video streams or the source device and is carried from Source to Sink across HDMI.
Multi-channel	Audio with more than 2 channels. Typically this term is applied to 6 (5.1) channel streams. Also called surround formats.
One Bit Audio	1-bit Delta-Sigma modulated signal stream such as that used by Super Audio CD.
Pixel	Picture Element. Refers to the actual element of the picture and the data in the digital video stream representing such an element.
Pixel Encoding	Bit placement and sequencing for the components of a pixel for a particular color space and chroma sampling.
Receiver	A component that is responsible for receiving the four differential TMDS input pairs at the input to an HDMI Sink and converting those signals into a digital output indicating a 24 bit, 12 bit, or 6 bit TMDS decoded word and indicating the TMDS coding mode used to decode those bits. This digital output may be contained within a semiconductor device or may be output from a semiconductor device.
(TMDS) Character	A 10-bit TMDS-encoded value. One such value is carried on each of the three data channels for each cycle of the TMDS clock.
Stereo	2 channel audio.
Stream	A time-ordered set of digital data originating from one Source and terminating at zero or more Sinks. A stream is characterized by bounded bandwidth requirements.
Super Audio CD	Disk format of “Super Audio CD System Description”, see http://www.licensing.philips.com.
T_{bit}	Time duration of a single bit carried across the TMDS data channels.

$T_{character}$	Time duration of a single TMDS character carried across the TMDS data channels. This is equal to $10*T_{bit}$.
Transmitter	A component that is responsible for driving the four differential TMDS output pairs into an HDMI output and for clocking the data driven into those four output pairs.
Video Field	The period from one VSYNC active edge to the next VSYNC active edge.
Video Format	A video format is sufficiently defined such that when it is received at the monitor, the monitor has enough information to properly display the video to the user. The definition of each format includes a Video Format Timing, the picture aspect ratio, and a colorimetry space.
Video Format Timing	The waveform associated with a video format. Note that a specific Video Format Timing may be associated with more than one Video Format (e.g., 720X480p@4:3 and 720X480p@16:9).
YC_BC_R	Digital representation of any video signal using one of several luminance/color-difference color spaces.

2.3 Acronyms and Abbreviations

ANSI	American National Standards Institute
AVI	Auxiliary Video Information
CEA	Consumer Electronics Association
CEC	Consumer Electronics Control
CTS	Cycle Time Stamp
DDC	Display Data Channel
DDWG	Digital Display Working Group
DST	Direct Stream Transport
DTD	Detailed Timing Descriptor
DTV	Digital Television
DVD	Digital Versatile Disc
DVI	Digital Visual Interface
E-DDC	Enhanced Display Data Channel
E-EDID	Enhanced Extended Display Identification Data
ECC	Error Correction Code

DELL/HP/LENOVO EXHIBIT 1015
Page 378

EDID	Extended Display Identification Data
EIA	Electronic Industries Alliance
HDCP	High-bandwidth Digital Content Protection
HDMI	High-Definition Multimedia Interface
HDTV	High-Definition Television
HPD	Hot Plug Detect
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
ITU	International Telecommunications Union
L-PCM	Linear Pulse-Code Modulation
LSb	least significant bit
MPEG	Moving Picture Experts Group
MSb	most significant bit
N.C.	No connect.
PCB	Printed Circuit Board
Rx	Receiver
SMPTE	Society of Motion Picture & Television Engineers
STB	Set-Top Box
SVD	Short Video Descriptor
TERC4	TMDS Error Reduction Coding – 4 bit
TMDS	Transition Minimized Differential Signaling
Tx	Transmitter
VESA	Video Electronics Standards Association
VSDB	Vendor-Specific Data Block

3 Overview

HDMI system architecture is defined to consist of Sources and Sinks. A given device may have one or more HDMI inputs and one or more HDMI outputs. Each HDMI input on these devices shall follow all of the rules for an HDMI Sink and each HDMI output shall follow all of the rules for an HDMI Source.

As shown in *Figure 3-1,* the HDMI cable and connectors carry four differential pairs that make up the TMDS data and clock channels. These channels are used to carry video, audio and auxiliary data. In addition, HDMI carries a VESA DDC channel. The DDC is used for configuration and status exchange between a single Source and a single Sink. The optional CEC protocol provides high-level control functions between all of the various audiovisual products in a user's environment.

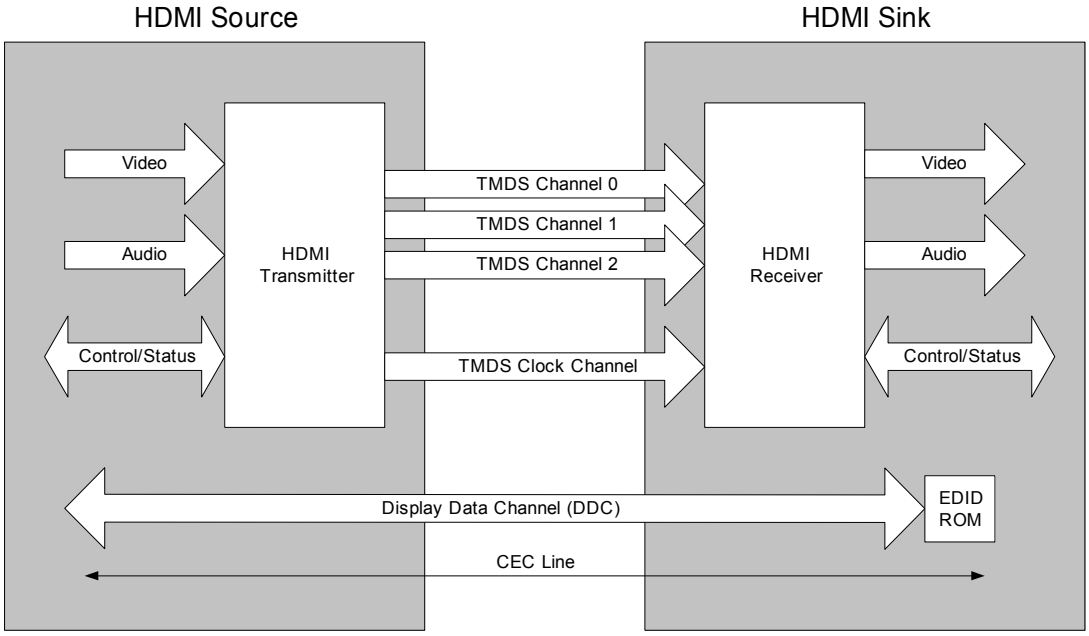


Figure 3-1 HDMI Block Diagram

Audio, video and auxiliary data is transmitted across the three TMDS data channels. A TMDS clock, typically running at the video pixel rate, is transmitted on the TMDS clock channel and is used by the receiver as a frequency reference for data recovery on the three TMDS data channels. At the source, TMDS encoding converts the 8 bits per TMDS data channel into the 10 bit DC-balanced, transition minimized sequence which is then transmitted serially across the pair at a rate of 10 bits per TMDS clock period.

Video data can have a pixel size of 24, 30, 36 or 48 bits. Video at the default 24-bit color depth is carried at a TMDS clock rate equal to the pixel clock rate. Higher color depths are carried using a correspondingly higher TMDS clock rate. Video formats with TMDS rates below 25MHz (e.g. 13.5MHz for 480i/NTSC) can be transmitted using a pixel-repetition scheme. The video pixels can be encoded in either RGB, YC_BC_R 4:4:4 or YC_BC_R 4:2:2 formats.

In order to transmit audio and auxiliary data across the TMDS channels, HDMI uses a packet structure. In order to attain the higher reliability required of audio and control data, this data is

protected with a BCH error correction code and is encoded using a special error reduction coding to produce the 10-bit word that is transmitted.

Basic audio functionality consists of a single IEC 60958 L-PCM audio stream at sample rates of 32kHz, 44.1kHz or 48kHz. This can accommodate any normal stereo stream. Optionally, HDMI can carry such audio at sample rates up to 192KHz and with 3 to 8 audio channels. HDMI can also carry an IEC 61937 compressed (e.g. surround-sound) audio stream at bit rates up to 24.576Mbps. HDMI can also carry from 2 to 8 channels of One Bit Audio and a compressed form of One Bit Audio called DST.

The DDC is used by the Source to read the Sink's Enhanced Extended Display Identification Data (E-EDID) in order to discover the Sink's configuration and/or capabilities.

4 Physical Layer

4.1 Connectors and Cables

4.1.1 Overview of Connectors

A device's external HDMI connection shall be presented via one of the three specified HDMI connectors, Type A, Type B or Type C. This connector can be attached directly to the device or can be attached via a cable adapter that is shipped with the device.

All three connectors carry all required HDMI signals, including a TMDS link. The Type B connector is slightly larger and carries a second TMDS link, which is necessary to support very high resolution displays using dual link. The Type C connector carries the same signals as the Type A but is more compact and intended for mobile applications.

Passive cable adapters between connector types are specified.

4.1.2 Connector Support Requirements

All features and functions are equally available to all three connectors.

4.1.3 Dual-Link

To support DVI signals greater than 165Mpixels/sec, the dual-link capability of the Type B connector shall be used. To support DVI signals less than or equal to 165Mpixels/sec, single-link operation shall be used.

To support very high-speed HDMI signals, the dual-link capability of the Type B connector is available. The single-link to dual-link crossover frequency for HDMI will be defined in a future specification and will be greater than 340Mpixels/sec. Dual-link cannot be used for formats below that crossover frequency.

4.1.4 Connector Pin Assignments

Table 4-1 Type A Connector Pin Assignment

PIN	Signal Assignment
1	TMDS Data2+
3	TMDS Data2–
5	TMDS Data1 Shield
7	TMDS Data0+
9	TMDS Data0–
11	TMDS Clock Shield
13	CEC
15	SCL
17	DDC/CEC Ground
19	Hot Plug Detect

PIN	Signal Assignment
2	TMDS Data2 Shield
4	TMDS Data1+
6	TMDS Data1–
8	TMDS Data0 Shield
10	TMDS Clock+
12	TMDS Clock–
14	Reserved (N.C. on device)
16	SDA
18	+5V Power

Table 4-2 Type B Connector Pin Assignment

PIN	Signal Assignment
1	TMDS Data2+
3	TMDS Data2-
5	TMDS Data1 Shield
7	TMDS Data0+
9	TMDS Data0-
11	TMDS Clock Shield
13	TMDS Data5+
15	TMDS Data5-
17	TMDS Data4 Shield
19	TMDS Data3+
21	TMDS Data3-
23	Reserved (N.C. on device)
25	SCL
27	DDC/CEC Ground
29	Hot Plug Detect

PIN	Signal Assignment
2	TMDS Data2 Shield
4	TMDS Data1+
6	TMDS Data1-
8	TMDS Data0 Shield
10	TMDS Clock+
12	TMDS Clock-
14	TMDS Data5 Shield
16	TMDS Data4+
18	TMDS Data4-
20	TMDS Data3 Shield
22	CEC
24	Reserved (N.C. on device)
26	SDA
28	+5V Power

Table 4-3 Type C Connector Pin Assignment

PIN	Signal Assignment
1	TMDS Data2 Shield
2	TMDS Data2+
3	TMDS Data2-
4	TMDS Data1 Shield
5	TMDS Data1+
6	TMDS Data1-
7	TMDS Data0 Shield
8	TMDS Data0+
9	TMDS Data0-
10	TMDS Clock Shield
11	TMDS Clock+
12	TMDS Clock-
13	DDC/CEC Ground
14	CEC
15	SCL
16	SDA
17	Reserved
18	+5V Power
19	Hot Plug Detect

4.1.5 Contact sequence

Table 4-4 Connector Contact Sequence

Connection	Signals	
	Type A & C Connectors	Type B Connector
First Make	Connector shell	Connector shell
Second Make	Pins 1 - 17 and pin 19	Pins 1 - 27 and pin 29
Third Make	Pin18 (+5V Power)	Pin28 (+5V Power)

4.1.6 Connector Mechanical Performance

Table 4-5 Type A and Type C Plug and Receptacle Mechanical Performance

Item	Test Condition	Requirement	
Vibration	Amplitude : 1.52mm P-P or 147m/s^2 {15G} Sweep time: 50-2000-50Hz in 20 minutes. Duration : 12 times in each (total of 36 Times) X, Y, Z axes. Electrical load : DC100mA current shall be Flowed during the test. (ANSI/EIA-364-28 Condition III)	Appearance	No Damage
		Contact Resistance	Contact : Change from initial value: 30 milliohms maximum. Shell Part : Change from initial value: 50 milliohms maximum.
		Discontinuity	1 μsec maximum.
Shock	Pulse width: 11 msec., Waveform : half sine, 490m/s^2{50G}, 3 strokes in each X.Y.Z. axes (ANSI/EIA-364-27, Condition A)	Appearance	No Damage
		Contact Resistance	Contact : Change from initial value: 30 milliohms maximum. Shell : Change from initial value: 50 milliohms maximum.
		Discontinuity	1 μsec maximum.
Durability	Measure contact and shell resistance after Following. Automatic cycling : Type A: 10,000 cycles at 100 ± 50 cycles per hour Type C: 5,000 cycles at 100 ± 50 cycles per hour	Contact Resistance	Contact : Change from initial value: 30 milliohms maximum. Shell : Change from initial value: 50 milliohms maximum.

Item	Test Condition	Requirement	
Insertion / Withdrawal Force	Insertion and withdrawal speed : 25mm/minute. (ANSI/EIA-364-13)	Withdrawal force	Type A: 9.8N {1.0kgf} minimum 39.2N {4.0kgf} maximum Type C: 7N minimum 25N maximum
		Insertion force	44.1N {4.5kgf} maximum
Cable Flex	100 cycles in each of 2 planes Dimension X = 3.7 x Cable Diameter. (ANSI/EIA-364-41C, Condition I)	Discontinuity	1 μsec maximum.
		Dielectric Withstanding Voltage and Insulation Resistance	Conform to item of dielectric withstanding voltage and insulation resistance

4.1.7 Connector Electrical Characteristics

4.1.7.1 Electrical Performance

Table 4-6 Electrical Performance

Item	Test Condition	Requirement
Contact Resistance	Mated connectors, Contact : measure by dry circuit, 20 mVolts maximum.,10mA. Shell : measured by open circuit, 5 Volts maximum ,100mA. (ANSI/EIA-364-06B)	Initial Contact resistance excluding conductor resistance: 10 milliohms maximum . (Target design value)
Dielectric Strength	Unmated connectors, apply 500 Volts AC(RMS) between adjacent terminal or ground. Mated connector, apply 300 Volts AC(RMS.) between adjacent terminal and ground. (ANSI/EIA-364-20C, Method A)	No Breakdown
Insulation Resistance	Unmated connectors, apply 500 Volts DC between adjacent terminal or ground. (ANSI/EIA 364-21C)	100 megaohms minimum (unmated)
	Mated connectors, apply 150 Volts DC between adjacent terminal or ground.	10 megaohms minimum (mated)
Contact Current Rating	55 °C, maximum ambient 85 °C, maximum temperature change (ANSI/EIA-364-70A)	0.5 A minimum
Applied Voltage Rating	40 Volts AC (RMS.) continuous maximum, on any signal pin with respect to the shield.	No Breakdown
Electrostatic Discharge	Test unmated each connectors from 1 kVolt to 8 kVolts in 1 kVolt steps using 8mm ball probe. (IEC-801-2)	No evidence of Discharge to Contacts at 8 kVolts

Item	Test Condition	Requirement
TMDS Signals Time Domain Impedance	Rise time ≤ 200 psec (10%-90%). Signal to Ground pin ratio per HDMI designation. Differential Measurement Specimen Environment Impedance = 100 ohms differential Source-side receptacle connector mounted on a controlled impedance PCB fixture. (ANSI/EIA-364-108)	Connector Area : Type A: 100 ohms ±15% Type C: 100 ohms ±25% Transition Area : 100 ohms ±15% Cable Area : 100 ohms ±10%
TMDS Signals Time Domain Cross talk FEXT	Rise time ≤ 200 psec (10%-90%). Signal to Ground pin ratio per HDMI designation. Differential Measurement Specimen Environment Impedance = 100 ohms differential. Source-side receptacle connector mounted on controlled impedance PCB fixture. Driven pair and victim pair. (ANSI/EIA-364-90)	Type A: 5% maximum Type C: 10% maximum

4.1.8 Connector Environmental Characteristics

4.1.8.1 Environmental Performance

Table 4-7 Connector Environmental Performance

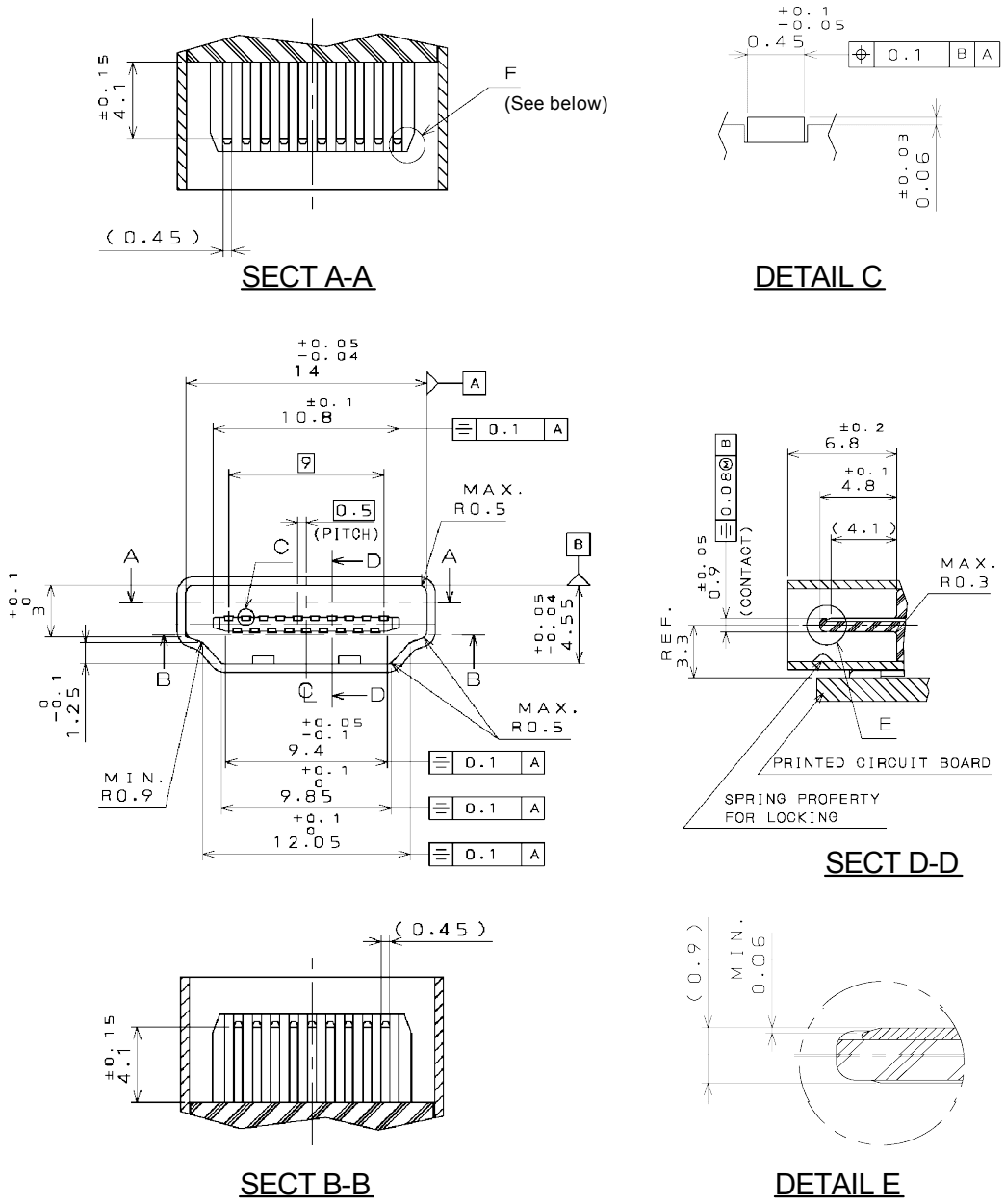
Item	Test Condition		Requirement	
Thermal Shock	10 cycles of: a) -55°Cfor 30 minutes b) +85°C for 30 minutes (ANSI/EIA-364-32C, Condition I)		Appearance	No Damage
			Contact Resistance	Contact : Change from initial value: 30 milliohms maximum. Shell Part : Change from initial value: 50 milliohms maximum.
Humidity	A	Mate connectors together and perform the test as follows. Temperature : +25 to +85°C Relative Humidity : 80 to 95% Duration : 4 cycles (96 hours) Upon completion of the test, specimens shall be conditioned at ambient room conditions for 24 hours, after which the specified measurements shall be performed. (ANSI/EIA-364-31B)	Appearance	No Damage
			Contact Resistance	Contact : Change from initial value: 30 milliohms maximum. Shell : Change from initial value: 50 milliohms maximum.
	B	Unmated each connectors and perform the test as follows. Temperature : +25 to +85°C Relative Humidity : 80 to 95% Duration : 4 cycles (96 hours) Upon completion of the test, specimens shall be conditioned at ambient room conditions for 24 hours, after which the specified measurements shall be performed. (ANSI/EIA-364-31B)	Appearance	No Damage
			Dielectric Withstanding Voltage and Insulation Resistance	Conform to item of Dielectric Withstanding Voltage and Insulation Resistance

Item	Test Condition	Requirement	
Thermal Aging	Mate connectors and expose to +105 ± 2°C for 250 hours. Upon completion of the exposure period, the test specimens shall be conditioned at ambient room conditions for 1 to 2 hours, after which the specified measurements shall be performed. (ANSI/EIA-364-17B, Condition 4, Method A)	Appearance	No Damage
		Contact Resistance	Contact : Change from initial value: 30 milliohms maximum. Shell Part : Change from initial value: 50 milliohms maximum.

4.1.9 Connector Drawings – Mating Interface Dimensions

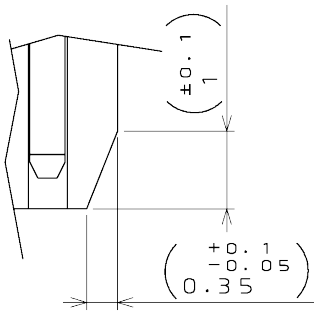
All dimensions in millimeters.

4.1.9.1 Type A Receptacle



- The shell shall have springs for locking. Additional springs may be used for EMI reduction.
- The spring property for locking shall be activated by the locking hole of the plug shell.

Figure 4-1 Type A Receptacle Mating Interface Dimensions

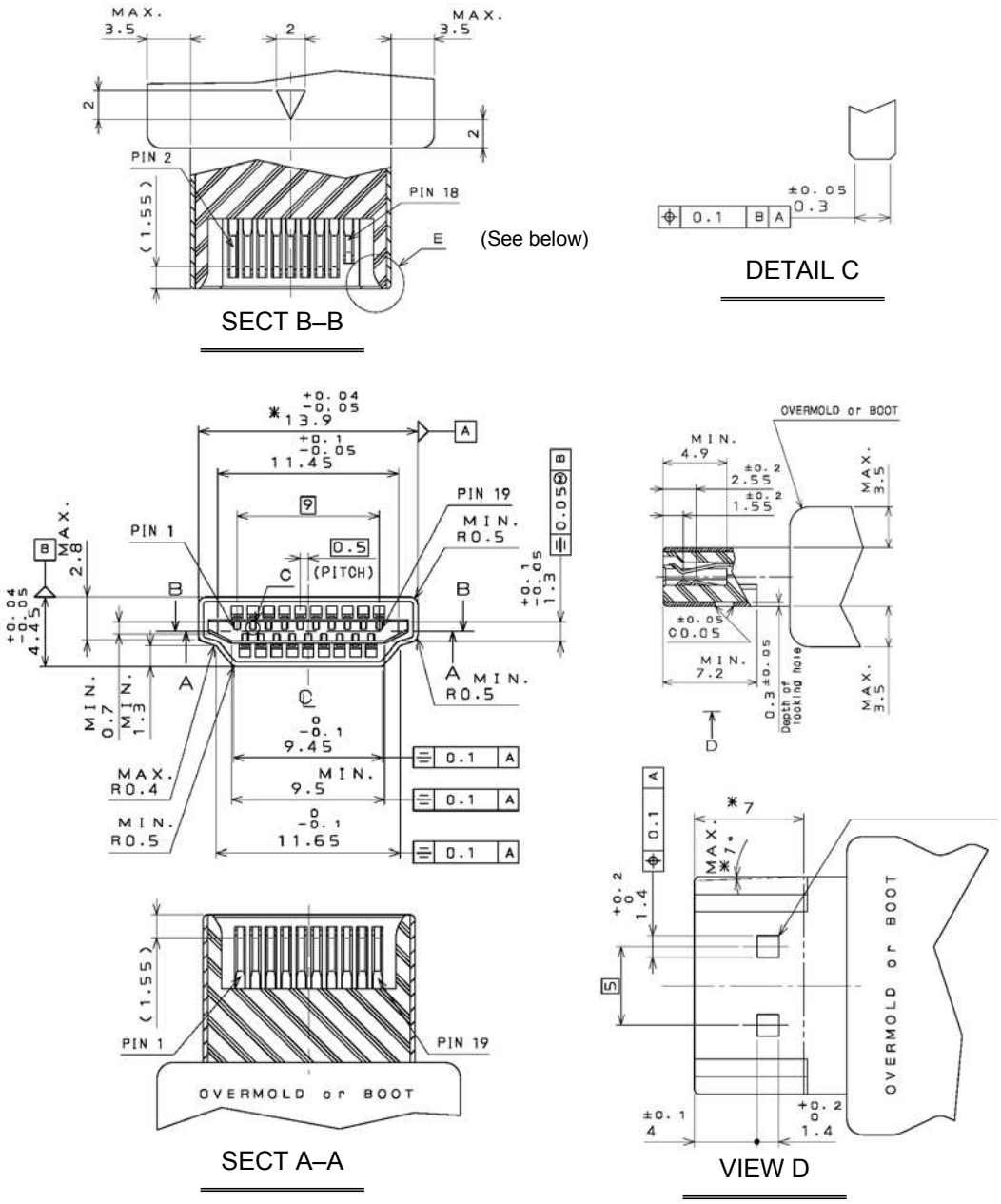


DETAIL F

- The form shown above is required. This feature will reduce the likelihood of damage to the receptacle insulator under rough operation.

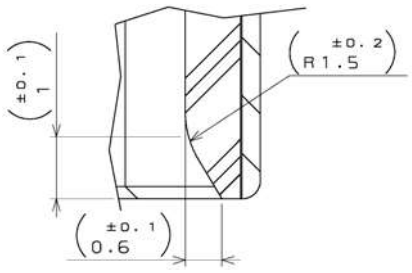
Figure 4-1–continued; Type A Receptacle, Detail F

4.1.9.2 Type A Plug



- The dimension of *13.9mm (+0.04 / -0.05) (on main section) should be measured at the point *7mm (on view D). The taper (on view D) shall be one degree max.
- The shell should not have a dimple other than the ones for locking.

Figure 4-2 Type A Plug Mating Interface Dimensions



DETAIL E

- The form shown above is required. This feature will reduce the likelihood of damage to the receptacle insulator under rough operation.

Figure 4-2-continued; Type A Plug, Detail E

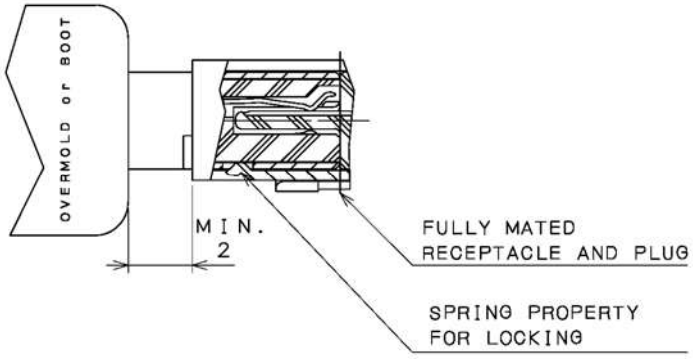


Figure 4-3 Type A Receptacle and Plug Mated Condition

It is recommended that products using Type A connectors be designed to ensure that cable bends are not tighter than that shown in Figure 4-4.

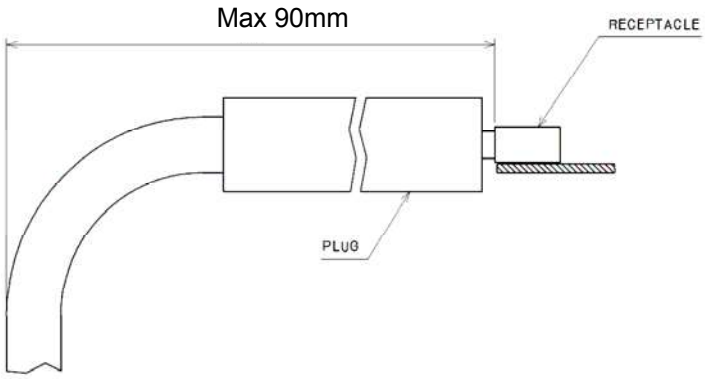
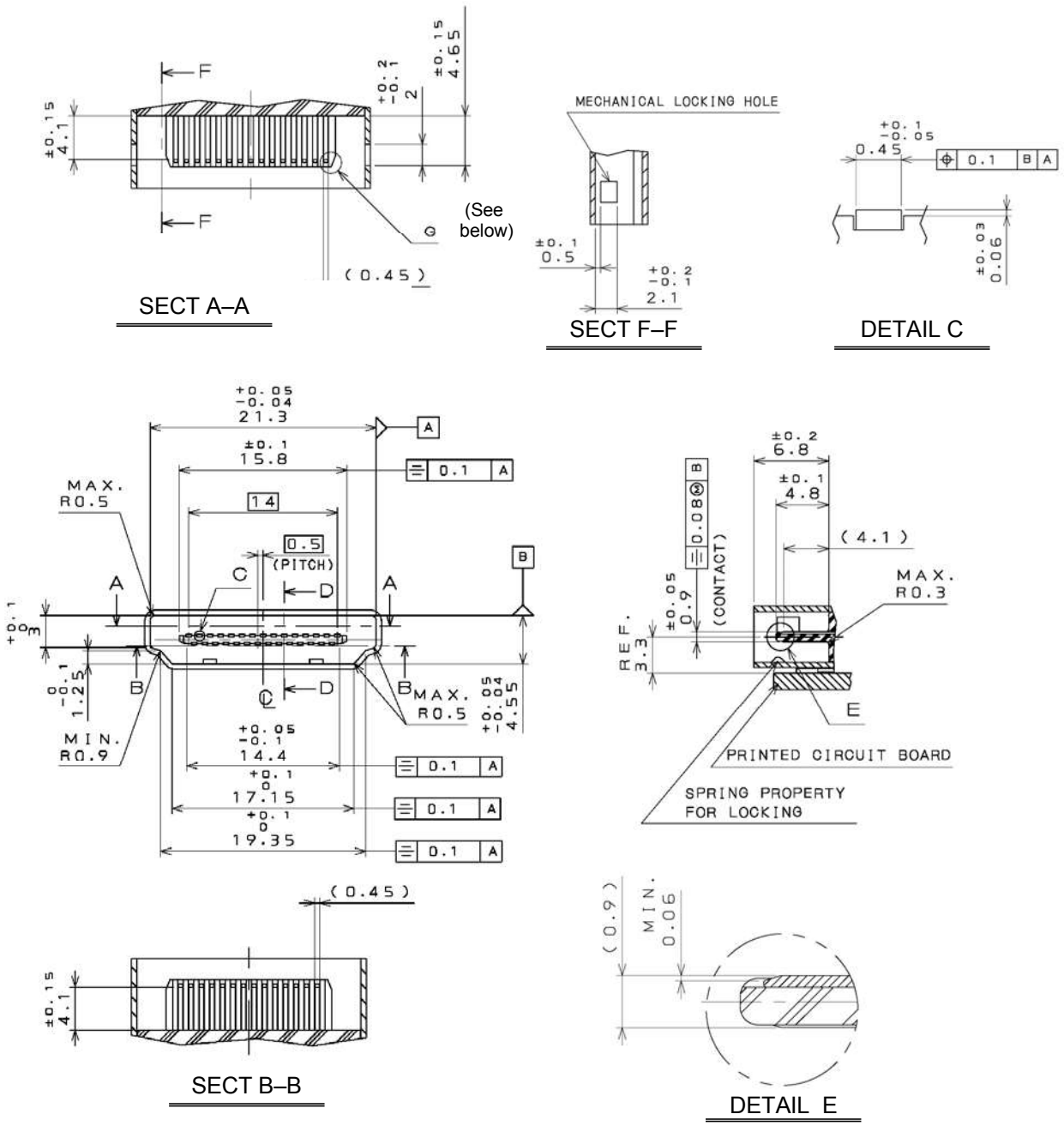


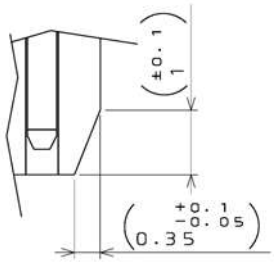
Figure 4-4 Type A Cable – Minimum Recommended Cable Bend

4.1.9.3 Type B Receptacle



- The shell shall have springs for locking. Additional springs may be used for EMI reduction.
- The spring property for locking shall be activated by the locking hole of the plug shell.

Figure 4-5 Type B Receptacle Mating Interface Dimensions

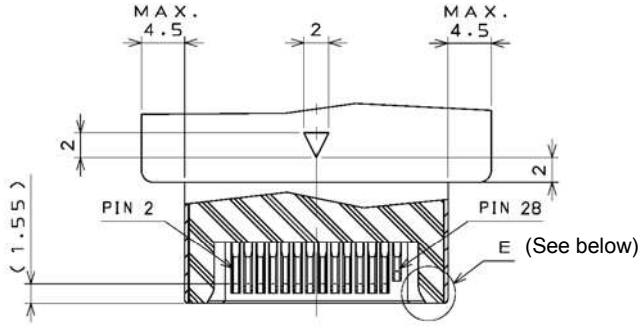


DETAIL G

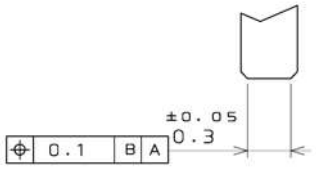
- The form shown above is required. This feature will reduce the likelihood of damage to the receptacle insulator under rough operation.

Figure 4-5-continued; Type B Receptacle, Detail G

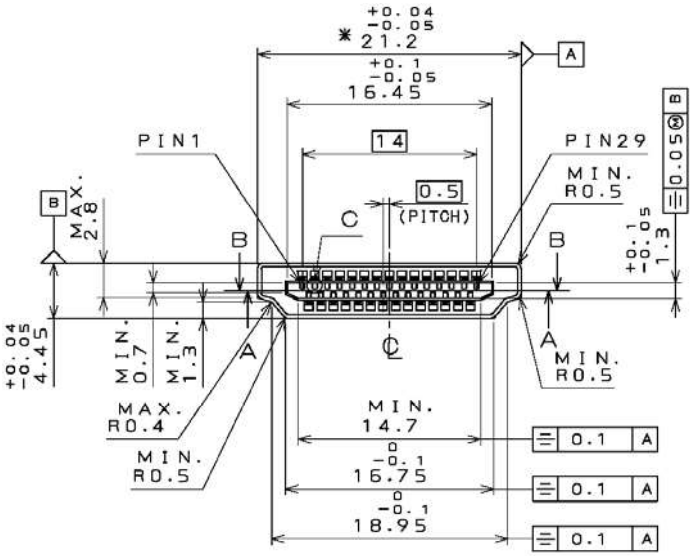
4.1.9.4 Type B Plug

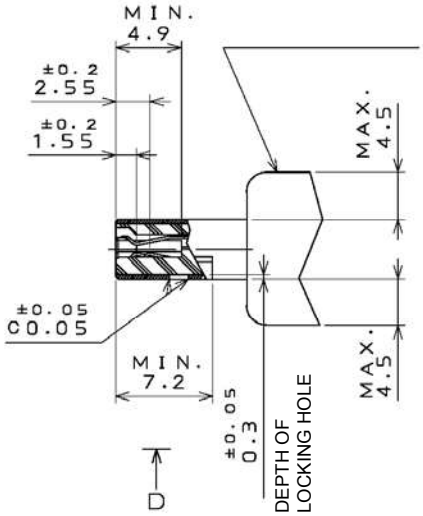


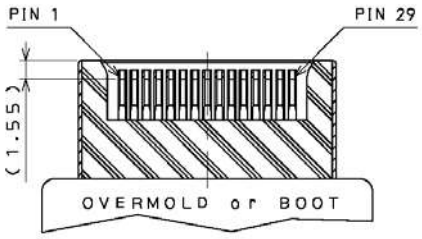
SECT B–B



DETAIL C



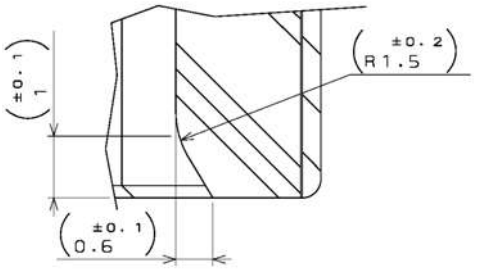




SECT A–A

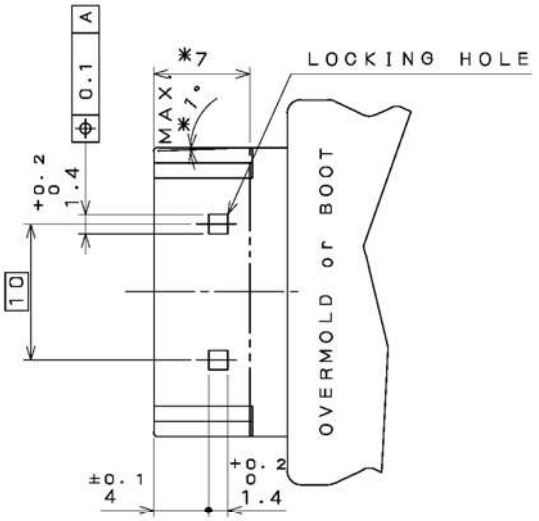
"VIEW D" → NEXT PAGE

- The dimension of *21.2mm (+0.04 / -0.05) (on main section) should be measured at the point *7mm (on view D). The taper (on view D) shall be one degree max.
- The shell should not have a dimple other than the ones for locking.

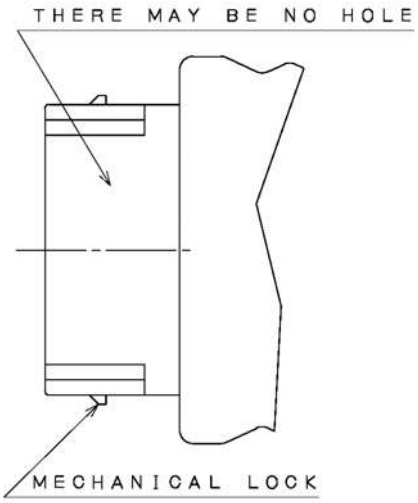


DETAIL E

- The form shown above is required. This feature will reduce the likelihood of damage to the receptacle insulator under rough operation.



FRICTION LOCK TYPE



MECHANICAL LOCK TYPE
The spring property for locking should be activated by the locking hole of the plug shell.

VIEW D

Figure 4-6 Type B Plug Mating Interface Dimensions

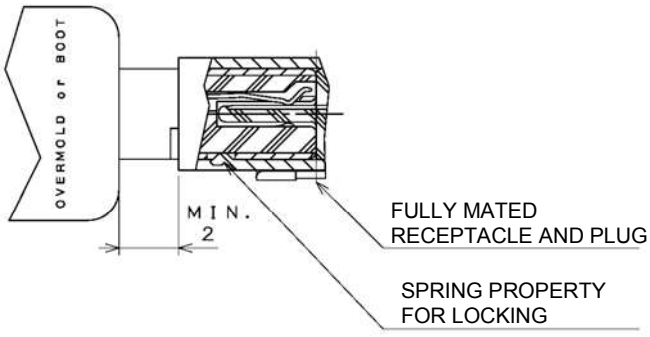
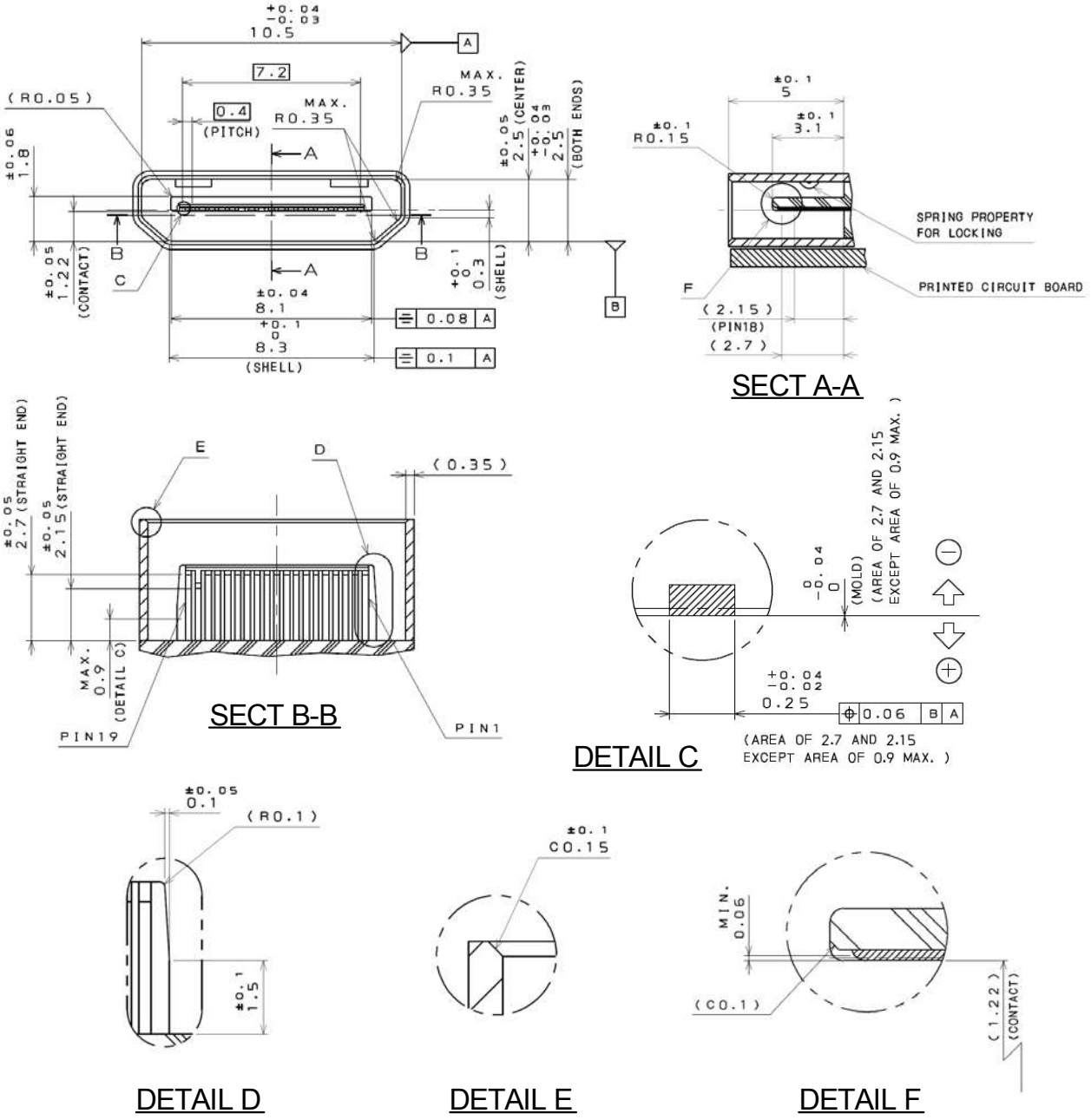


Figure 4-7 Type B Receptacle and Plug Mated Condition

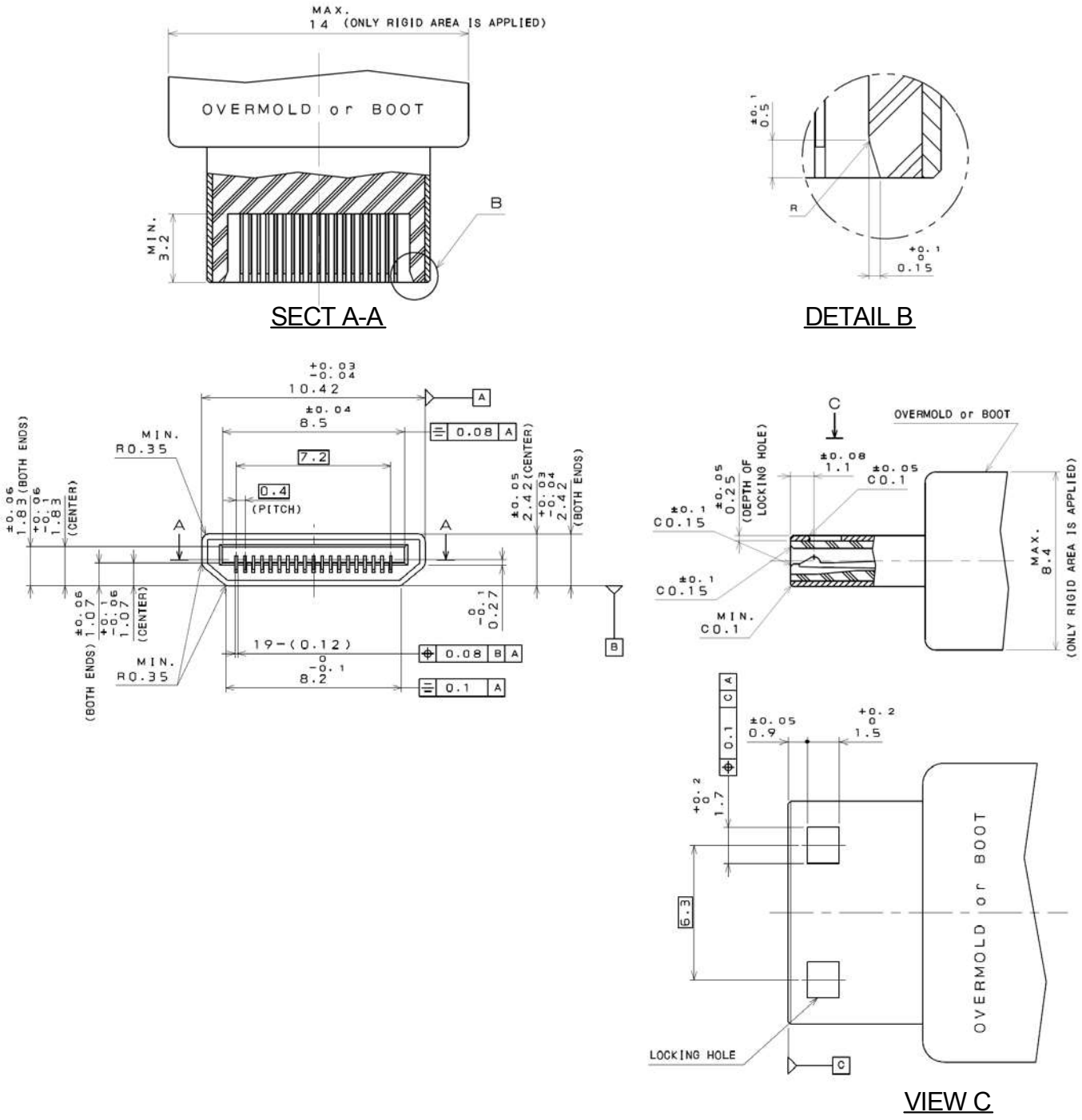
4.1.9.5 Type C Receptacle



- The shell shall have spring for locking. Additional springs may be used for EMI reduction
- The spring property for locking shall be activated by the locking hole of the plug shell.

Figure 4-8 Type C Receptacle Mating Interface Dimensions

4.1.9.6 Type C Plug



- The shell should not have a dimple other than the ones for locking.
- The cable should have a maximum diameter of 7mm

Figure 4-9 Type C Plug Mating Interface Dimensions

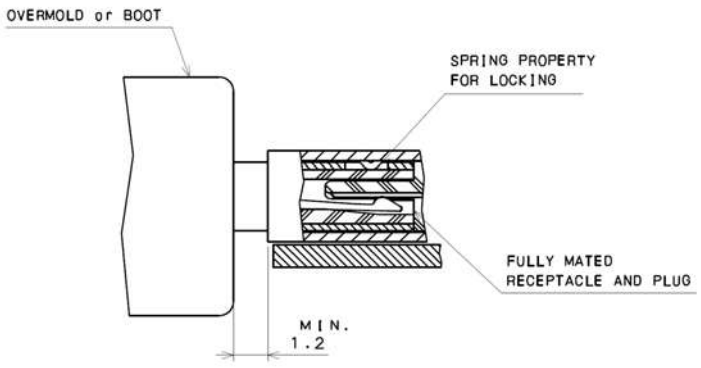


Figure 4-10 Type C Receptacle and Plug Mated Condition

It is recommended that products using Type C connectors be designed to ensure that cable bends that are not tighter than that shown in Figure 4-11. In addition, for strength, it is recommended that the shell flange and the case be fixed with a screw and that the clearance between the connector and the case be as narrow as possible.

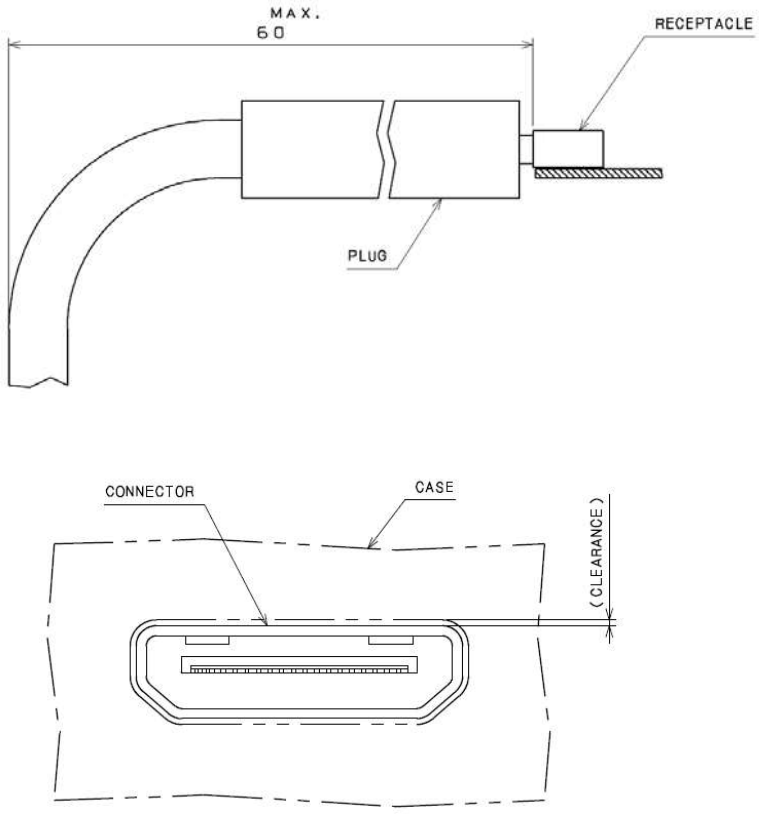


Figure 4-11 Type C Product Design Recommendations

4.1.10 Cable Adapter Specification

Table 4-8 Wire Categories

Category	Description
A	TMDS Signal Wire
B	TMDS Shield
C	Control
D	Control Ground
N.C.	No connect (no wire)
5V	5 Volts Power Wire

4.1.10.1 Type A Connector to Type A Connector

Table 4-9 Type A-to-Type A Cable Wire Assignment

Type A pin	Signal Name	Wire	Type A pin
1	TMDS Data2+	A	1
2	TMDS Data2 Shield	B	2
3	TMDS Data2–	A	3
4	TMDS Data1+	A	4
5	TMDS Data1 Shield	B	5
6	TMDS Data1–	A	6
7	TMDS Data0+	A	7
8	TMDS Data0 Shield	B	8
9	TMDS Data0–	A	9
10	TMDS Clock+	A	10
11	TMDS Clock Shield	B	11
12	TMDS Clock–	A	12
13	CEC	C	13
14	Reserved (in cable but N.C. on device)	C	14
15	SCL	C	15
16	SDA	C	16
17	DDC/CEC Ground	D	17
18	+5V Power	5V	18
19	Hot Plug Detect	C	19

4.1.10.2 Type A Connector to Type B Connector

Table 4-10 Type A-to-Type B Cable Wire Assignment

Type A pin	Pin Assignment	Wire	Type B pin
1	TMDS Data2+	A	1
2	TMDS Data2 Shield	B	2
3	TMDS Data2-	A	3
4	TMDS Data1+	A	4
5	TMDS Data1 Shield	B	5
6	TMDS Data1-	A	6
7	TMDS Data0+	A	7
8	TMDS Data0 Shield	B	8
9	TMDS Data0-	A	9
10	TMDS Clock+	A	10
11	TMDS Clock Shield	B	11
12	TMDS Clock-	A	12
13	CEC	C	22
15	SCL	C	25
16	SDA	C	26
17	DDC/CEC Ground	D	27
18	+5V Power	5V	28
19	Hot Plug Detect	C	29
14	No connect	N.C.	
	No connect	N.C.	23
	No connect	N.C.	24

4.1.10.3 Type B Connector to Type B Connector

Table 4-11 Type B to Type B Cable Wire Assignment

Type B pin	Pin Assignment	Wire	Type B pin
1	TMDS Data2+	A	1
2	TMDS Data2 Shield	B	2
3	TMDS Data2-	A	3
4	TMDS Data1+	A	4
5	TMDS Data1 Shield	B	5
6	TMDS Data1-	A	6
7	TMDS Data0+	A	7
8	TMDS Data0 Shield	B	8
9	TMDS Data0-	A	9
10	TMDS Clock+	A	10
11	TMDS Clock Shield	B	11
12	TMDS Clock-	A	12
13	TMDS Data5+	A	13
14	TMDS Data5 Shield	B	14
15	TMDS Data5-	A	15
16	TMDS Data4+	A	16
17	TMDS Data4 Shield	B	17
18	TMDS Data4-	A	18
19	TMDS Data3+	A	19
20	TMDS Data3 Shield	B	20
21	TMDS Data3-	A	21
22	CEC	C	22
25	SCL	C	25
26	SDA	C	26
27	DDC/CEC Ground	D	27
28	+5V Power	5V	28
29	Hot Plug Detect	C	29
23	No Connect	N.C.	
24	No Connect	N.C.	
	No Connect	N.C.	23
	No Connect	N.C.	24

DELL/HP/LENOVO EXHIBIT 1015
Page 407

4.1.10.4 Type C Connector to Type C Connector

Table 4-12 Type C-to-Type C Cable Wire Assignment

Type C pin	Signal Name	Wire	Type C pin
1	TMDS Data2 Shield	B	1
2	TMDS Data2+	A	2
3	TMDS Data2-	A	3
4	TMDS Data1 Shield	B	4
5	TMDS Data1+	A	5
6	TMDS Data1-	A	6
7	TMDS Data0 Shield	B	7
8	TMDS Data0+	A	8
9	TMDS Data0-	A	9
10	TMDS Clock Shield	B	10
11	TMDS Clock+	A	11
12	TMDS Clock-	A	12
13	DDC/CEC Ground	D	13
14	CEC	C	14
15	SCL	C	15
16	SDA	C	16
17	Reserved (in cable but N.C. on device)	C	17
18	+5V Power	5V	18
19	Hot Plug Detect	C	19

4.1.10.5 Type C Connector to Type A Connector

Table 4-13 Type C-to-Type A Cable Wire Assignment

Type C pin	Signal Name	Wire	Type A pin
1	TMDS Data2 Shield	B	2
2	TMDS Data2+	A	1
3	TMDS Data2-	A	3
4	TMDS Data1 Shield	B	5
5	TMDS Data1+	A	4
6	TMDS Data1-	A	6
7	TMDS Data0 Shield	B	8
8	TMDS Data0+	A	7
9	TMDS Data0-	A	9
10	TMDS Clock Shield	B	11
11	TMDS Clock+	A	10
12	TMDS Clock-	A	12
13	DDC/CEC Ground	D	17
14	CEC	C	13
15	SCL	C	15
16	SDA	C	16
17	Reserved (in cable but N.C. on device)	C	14
18	+5V Power	5V	18
19	Hot Plug Detect	C	19

4.2 Electrical Specification

Some timing parameter values in this specification are based on the clock rate of the link while others are based on absolute values. For scalable timing parameters based on the TMDS clock rate, the time period of the clock is denoted as 'TMDS character time', or $T_{character}$. One tenth of the character time is called the bit time, or T_{bit}. The bit time is also referred to as one Unit Interval in the jitter and eye diagram specifications.

Schematic diagrams contained in this chapter are for illustration only and do not represent the only feasible implementation.

4.2.1 TMDS Overview

The conceptual schematic of one TMDS differential pair is shown in Figure 4-12. TMDS technology uses current drive to develop the low voltage differential signal at the Sink side of the DC-coupled transmission line. The link reference voltage AV_{cc} sets the high voltage level of the differential signal, while the low voltage level is determined by the current source of the HDMI Source and the termination resistance at the Sink. The termination resistance (R_T) and the characteristic impedance of the cable (Z_0) must be matched.

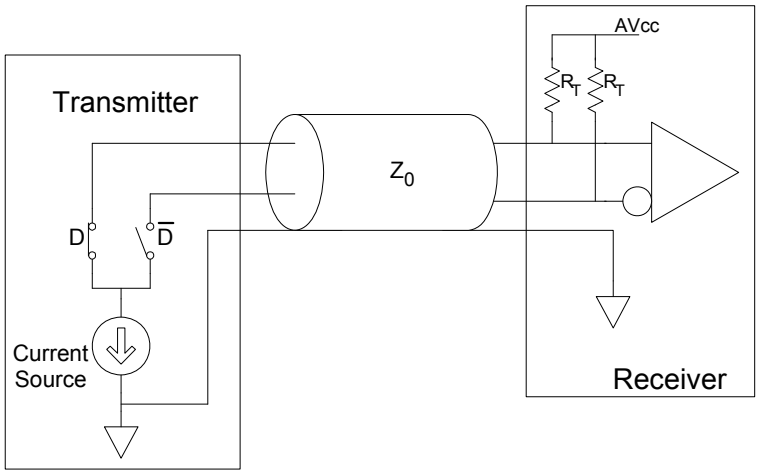


Figure 4-12 Conceptual Schematic for one TMDS differential pair

A single-ended differential signal, representing either the positive or negative terminal of a differential pair, is illustrated in Figure 4-13. The nominal high-level voltage of the signal is AV_{cc} and the nominal low-level voltage of the signal is (AV_{cc} - V_{swing}). Since the swing is differential on the pair, the net signal on the pair has a swing twice that of the single-ended signal, or $2*V_{swing}$. The differential signal, as shown in Figure 4-14, swings between positive V_{swing} and negative V_{swing}.

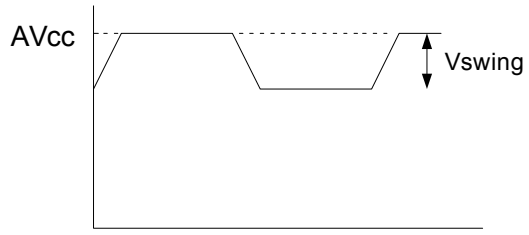


Figure 4-13 Single-ended Differential Signal

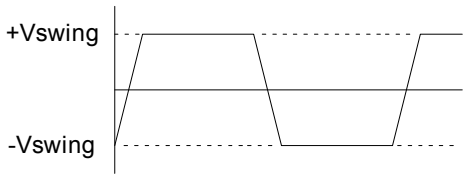


Figure 4-14 Differential Signal

The signal test points for a TMDS link are shown in Figure 4-15. TP1 is used for testing of HDMI Sources and Transmitter components. TP2 is used for testing of HDMI Sinks and Receiver components. TP1 and TP2 together are also used for testing of cables.

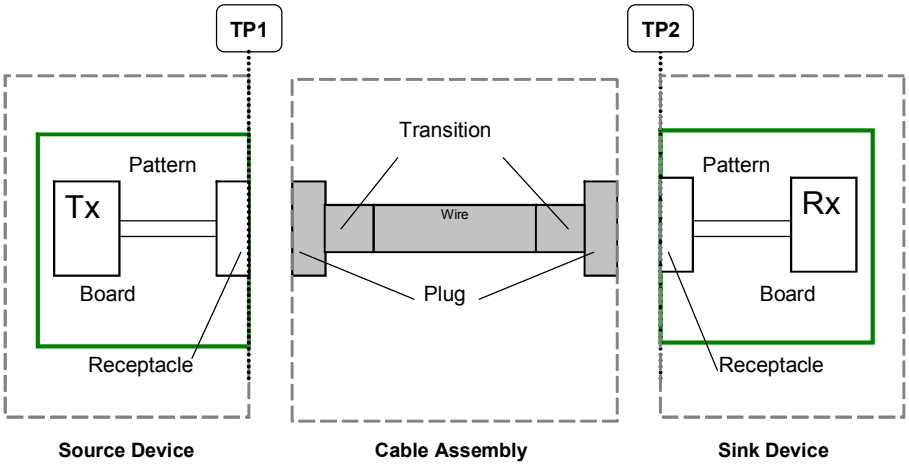


Figure 4-15 TMDS Link Test Points

4.2.2 TMDS System Operating Conditions

The required operating conditions of the TMDS pairs are specified in Table 4-14.

Table 4-14 Required Operating Conditions for HDMI Interface (see Figure 4-12)

Item	Value
Termination Supply Voltage, AV_{cc}	3.3 Volts ±5%
Termination Resistance, R_T	50 ohms ±10%

4.2.3 TMDS Specification and Testing Overview

4.2.3.1 Jitter and Eye Measurements: Ideal Recovery Clock

All TMDS Clock and Data signal jitter specifications are specified relative to an Ideal Recovery Clock defined below. The Data jitter is not specified numerically, but instead, an HDMI device or cable shall adhere to the appropriate eye diagram(s) when the TMDS data signals are measured using an Ideal Recovery Clock as a trigger source.

The TMDS Clock signal may contain low-frequency jitter components, which can be tracked by a Sink's clock recovery circuitry, and high-frequency components, which are not typically tracked. The purpose of the Ideal Recovery Clock is to give an accurate representation of link performance when used as a trigger for eye diagram and clock jitter specifications.

For the purposes of jitter and eye diagram specification, the Ideal Recovery Clock is defined relative to the TMDS clock signal. The Ideal Recovery Clock shall be equivalent to the signal that would be derived by a perfect PLL (Ideal Clock Recovery Unit) with a jitter transfer function shown in Equation 4-1, when the TMDS clock signal were input into that PLL. This jitter transfer function has the behavior of a low pass filter with 20dB/decade roll-off and with a –3dB point of 4MHz.

For the purposes of compliance testing, a Clock Recovery Unit is used to generate a Recovered Clock, which is meant to approximate the Ideal Recovery Clock. This Recovered Clock is used for measurement of the jitter and eye diagram.

$$H(j\omega) = 1 / (1 + j\omega/\omega_0)$$

$$\text{Where } \omega_0 = 2\pi F_0, \; F_0 = 4.0\text{MHz}$$

Equation 4-1 Jitter Transfer Function of Ideal CRU for Ideal Recovery Clock Definition

4.2.3.2 Reference Cable Equalizer

The signal degradation introduced by typical passive cables increases with the frequency of the signal and the length of the cable. In order to accommodate passive copper cables of market-required lengths at the very high frequencies supported by HDMI, higher-speed HDMI Sinks are expected to support some sort of cable equalization function which allows them to recover data from such cables.

For lower-speed operation, the HDMI cable is specified with respect to the worst-case Source output eye (cable input eye) and the Sink input eye. For higher-speed operation, the HDMI cable

specification also assumes application by the Sink of a cable equalization function approximating the performance implied by the Reference Cable Equalizer, which is a specified mathematical model of cable equalization.

The HDMI Sink is required to successfully recover the data stream from any compliant Sink input signal. At high frequencies, a compliant Sink input signal is any signal that, after application of the Reference Cable Equalizer to each of the differential TMDS signals, results in a signal that meets the Sink input eye requirements.

The definition of the Reference Cable Equalizer is given in Equation 4-2. The gain of this equation is shown in Figure 4-16.

$$\left|H(j\omega)\right| = \begin{cases} e^{A*\omega^N} & (\omega < \omega_0) \\ e^{-B*(\omega-1.2*\omega_0)^2+C} & (\omega_0 < \omega < 1.4*\omega_0) \\ e^{-D*\omega+E} & (1.4*\omega_0 < \omega) \end{cases}$$

Where:

$$N = 0.7$$

$$\omega_0 = 2\pi * 2.25GHz$$

$$A = 7.34E-8$$

$$B = \frac{7}{4} * A * \omega_0^{-1.3}$$

$$C = 1.07 * A * \omega_0^{0.7}$$

$$D = 0.7 * A * \omega_0^{-0.3}$$

$$E = 1.98 * A * \omega_0^{0.7}$$

Equation 4-2 Reference Cable Equalizer Function

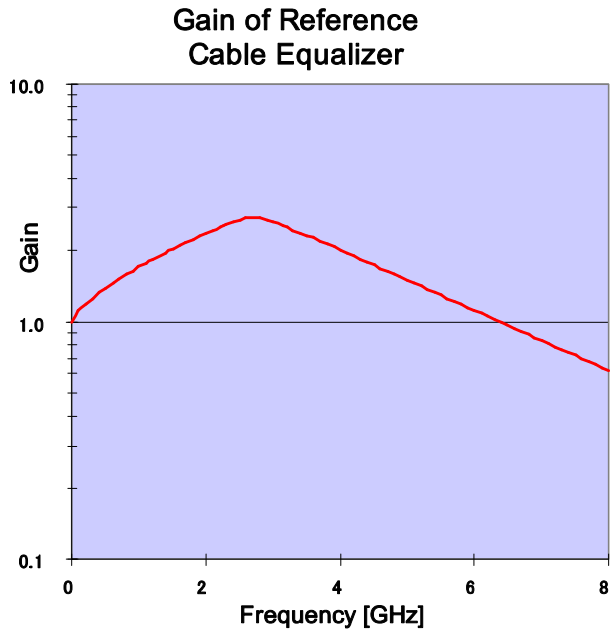


Figure 4-16 Gain of Reference Cable Equalizer

4.2.4 HDMI Source TMDS Characteristics

HDMI requires a DC-coupled TMDS link. Source electrical testing shall be performed using the test load shown in Figure 4-17, with AV_{cc} set to 3.3V. TP1 represents the connection point of the receptacle.

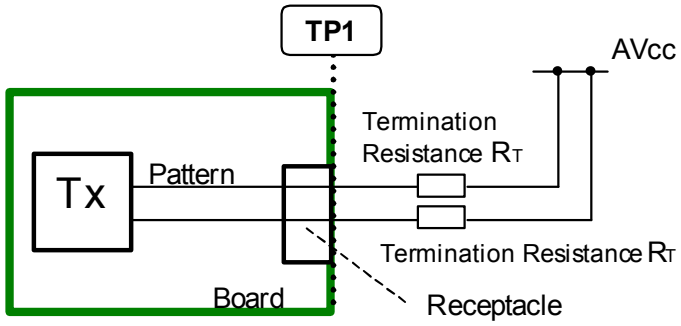


Figure 4-17 Balanced Source Test Load

The Source shall meet the DC specifications in Table 4-15 for all operating conditions specified in Table 4-14 when driving clock and data signals. The V_{swing} parameter is the difference between the single-ended most common high-level voltage (as would be revealed with a histogram measurement) and the most-common low-level voltage, after ringing has subsided.

Table 4-15 Source DC Characteristics at TP1

Item	Value
Single-ended standby (off) output voltage, V_{OFF}	AV_{cc} ±10mVolts
Single-ended output swing voltage, V_{swing}	400mVolts ≤ V_{swing} ≤ 600mVolts
Single-ended high level output voltage, V_H	if attached Sink supports only <=165MHz : AV_{cc} ±10mVolts if attached Sink supports >165MHz : (AV_{cc} – 200mVolts) ≤ V_H ≤ (AV_{cc} + 10mVolts)
Single-ended low level output voltage, V_L	if attached Sink supports only <=165MHz : (AV_{cc} – 600mVolts) ≤ V_L ≤ (AV_{cc} – 400mVolts) if attached Sink supports >165MHz : (AV_{cc} – 700mVolts) ≤ V_L ≤ (AV_{cc} – 400mVolts)

It is recommended that Sources capable of higher speeds incorporate some amount of source termination. This termination will typically have the effect of lowering the average DC level of each single-ended signal. The relaxed V_H and V_L parameters permit such an implementation.

The Source shall meet the AC specifications in Table 4-16 across all operating conditions specified in Table 4-14. Rise and fall times are defined as the signal transition time between 20% and 80% of the nominal swing voltage (V_{swing}) of the device under test.

The Source intra-pair skew is the maximum allowable time difference (on both low-to-high and high-to-low transitions) as measured at TP1, between the true and complement signals of a given differential pair. This time difference is measured at the midpoint on the single-ended signal swing of the true and complement signals. The Source inter-pair skew is the maximum allowable time difference (on both low-to-high and high-to-low transitions) as measured at TP1, between any two single-ended data signals that do not constitute a differential pair.

Table 4-16 Source AC Characteristics at TP1

Item	Value
Rise time / fall time (20%-80%)	75psec ≤ Rise time / fall time ≤ 0.4 T_{bit}
Undershoot, max	25% of full differential amplitude (V_{swing}*2)
Intra-Pair Skew at Source Connector, max	0.15 T_{bit}
Inter-Pair Skew at Source Connector, max	0.20 $T_{character}$
Clock duty cycle, min / average / max	40% / 50% / 60%
TMDS Differential Clock Jitter, max	0.25 T_{bit} (relative to Ideal Recovery Clock as defined in Section 4.2.3)

The design of a Source should take into account the differential impedance of the cable assembly and Sink of 100 ohms (see Table 4-21).

For all channels under all operating conditions specified in Table 4-14 and when terminated as shown in Figure 4-17, the Source shall have output levels at TP1that meet the normalized eye diagram requirements of Figure 4-18. This requirement, normalized in both time and amplitude, specifies the minimum eye opening relative to the average differential swing voltage as well as the absolute maximum and minimum voltages. The time axis is normalized to the bit time at the operating frequency, while the minimum eye amplitude is normalized to the average differential swing voltage.

Determination of average swing voltage is made with transmission of the half clock test pattern. The average high-level and low-level amplitudes are determined at the point where signal ringing has subsided. These averages establish the swing voltage and are used to normalize the eye diagram.

The average differential swing voltage is defined as the difference between the average differential amplitude when driving a logic one and the average differential amplitude when driving a logic zero. The average logic one appears at positive 0.5 on the vertical axis, while the average logic zero appears at negative 0.5. The normalized amplitude limits in Figure 4-18 allow 25% (of the average differential swing voltage) maximum undershoot, relative to the amplitudes determined to be logic one and zero. Overshoot limits are imposed only by the absolute max/min voltages of ±780mV shown above and below the normalized eye.

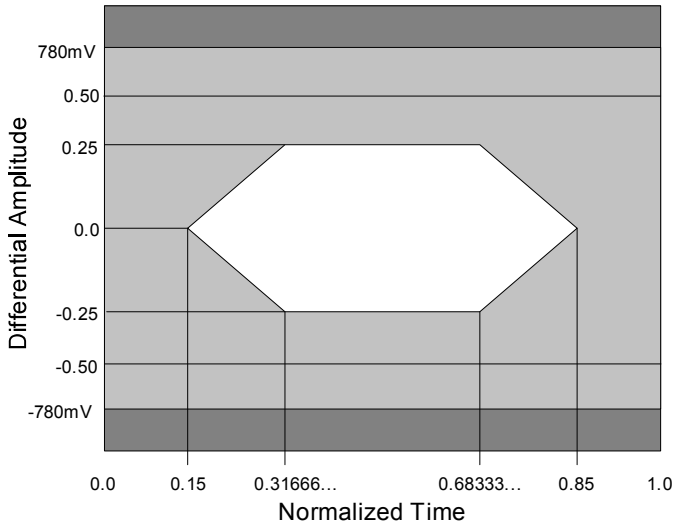


Figure 4-18 Eye Diagram Mask at TP1 for Source Requirements

Combining the single-ended swing voltage (V_{swing}) specified in Table 4-15 with a 15% overshoot (legacy) and the undershoot limits of Table 4-16, it is possible to calculate the minimum and maximum high-level voltage (V_{high}) and low-level voltage (V_{low}) that is allowable on the interface.

V_{high} (max) = V_{swing} (max) + 15% * (2*V_{swing} (max)) = 600 + 180 = 780 mV

V_{high} (min) = V_{swing} (min) - 25% * (2*V_{swing} (min)) = 400 - 200 = 200 mV

V_{low} (max) = -V_{swing} (max) - 15% * (2*V_{swing} (max)) = -600 - 180 = -780 mV

V_{low} (min) = -V_{swing} (min) + 25% * (2*V_{swing} (min)) = -400 + 200 = -200 mV

Minimum opening at Source = V_{high} (min) - V_{low} (min) = 400 mV

Note that the combination of these extreme cases do not constitute a single valid eye.

Source eye diagram test procedures are defined in the HDMI Compliance Test Specification. The Source eye diagram mask of Figure 4-18 is not used for response time and clock jitter specifications, but specifies the clock to data jitter indirectly.

4.2.5 HDMI Sink TMDS Characteristics

HDMI Sink electrical testing shall be performed using a test signal generator as shown in Figure 4-19.

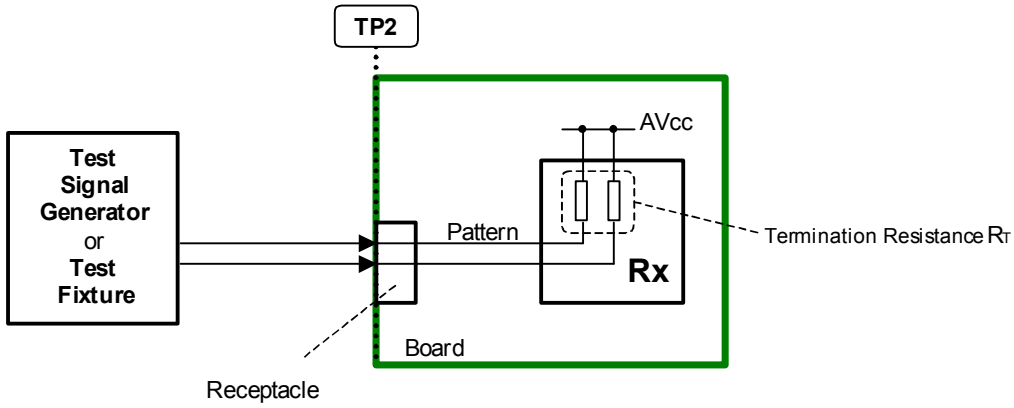


Figure 4-19 HDMI Sink Test Points

There may be a risk of source damage if the Sink asserts a very high or very low voltage, such as beyond the maximum ratings in the DVI 1.0 specification, on any TMDS line during power-on or other power transitions.

The Sink shall meet the signal requirements listed in Table 4-17, Table 4-18, and Table 4-19.

Table 4-17 Sink Operating DC Characteristics at TP2

Item		Value
Input Differential Voltage Level, V_{idiff}		$150 \le V_{idiff} \le 1200$ mVolts
Input Common Mode Voltage, V_{icm}	V_{icm1}	if Sink supports only <=165MHz : $(AV_{cc} - 300mVolts) \le V_{icm1} \le (AV_{cc} - 37.5mVolts)$ If Sink supports >165MHz : $(AV_{cc} - 400mVolts) \le V_{icm1} \le (AV_{cc} - 37.5mVolts)$
	V_{icm2}	AV_{cc} ±10mVolts

All Sinks are required to support both V_{icm} ranges (V_{icm1} and V_{icm2}). Sources are not yet permitted to operate in the V_{icm2} (AC-coupled) range. At higher speeds, Source devices may implement source termination, which may lower the DC-coupled V_{icm} (V_{icm1}) seen by the Sink.

Table 4-18 Sink DC Characteristics When Source Disabled or Disconnected at TP2

Item	Value
Differential Voltage Level	AV_{cc} ±10mVolts

Table 4-19 Sink AC Input Characteristics at TP2

Item	Value
Minimum differential sensitivity (peak-to-peak)	150 mVolts
Maximum differential input (peak-to-peak)	1560 mVolts
Max Allowable Intra-Pair Skew at Sink Connector	For TMDS Clock rates 225MHz and below: 0.4 T_{bit} For TMDS Clock rates above 225MHz: 0.15 T_{bit} + 111psecs
Max Allowable Inter-Pair Skew at Sink Connector	0.2 $T_{character}$ + 1.78nsecs
TMDS Clock Jitter	0.30 T_{bit} (relative to Ideal Recovery Clock as defined in Section 4.2.3, for all TMDS clock rates >165MHz: after application of Reference Cable Equalizer given in Equation 4-2)

For each channel under all operating conditions specified in this section the following conditions shall be met. At TMDS clock frequencies less than or equal to 165MHz, the Sink shall recover data at a TMDS character error rate of 10^{-9} or better, when presented with any signal compliant to the eye diagram of Figure 4-20. At TMDS clock frequencies above 165MHz, the Sink shall recover data on each channel at a TMDS character error rate of 10^{-9} or better, when presented with any signal compliant to the eye diagram of Figure 4-20 after application of the Reference Cable Equalizer.

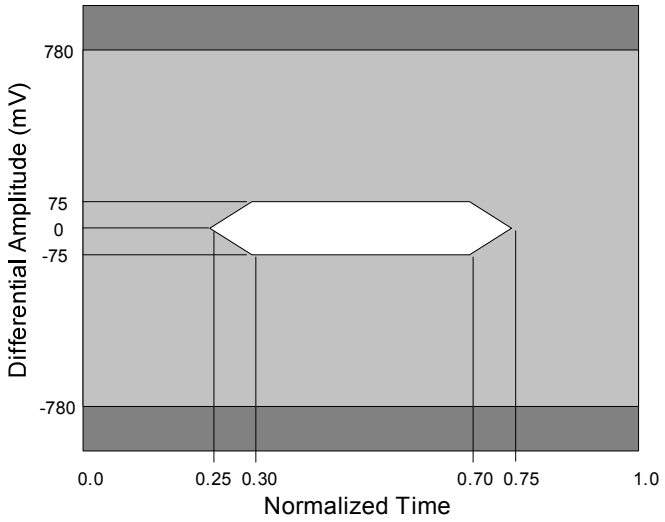


Figure 4-20 Eye Diagram Mask at TP2 for Sink Requirements

Table 4-20 HDMI Sink Impedance Characteristics at TP2

Item	Value
TDR Rise Time at TP2 (10%-90%)	≤200 psec
Through connection impedance	100 ohms ±15%
At Termination impedance (when V_{icm} is within V_{icm1} range)	100 ohms ±10%
At Termination impedance (when V_{icm} is within V_{icm2} range)	100 ohms ±35%

4.2.6 Cable Assembly TMDS Characteristics

The term “Cable assembly” includes all five parts listed below:

- Source-side plug
- Source-side transition (from plug to cable)
- Cable itself
- Sink-side transition
- Sink-side plug

HDMI cables are measured with respect to the test points TP3 and TP4 shown in Figure 4-21. TP1 and TP2 are not available because connection points between plug and receptacle cannot be accessed during testing. Therefore, TP3 and TP4 are used, even though the effects of receptacles at both ends are included in the test result.

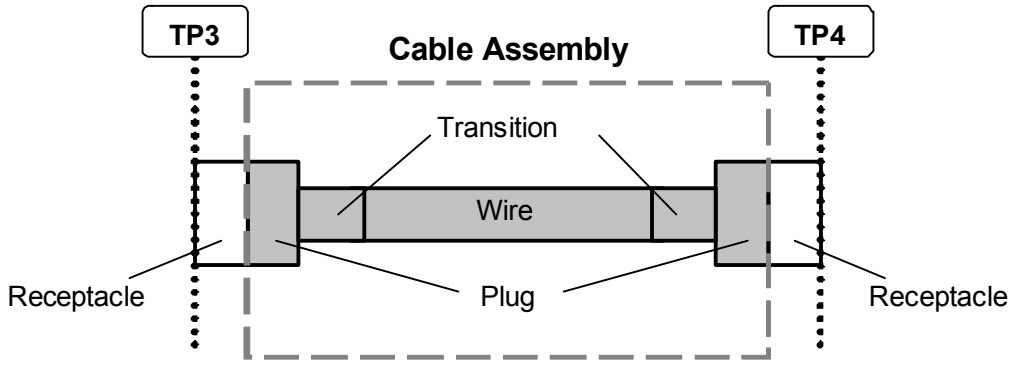


Figure 4-21 Cable Assembly Test Points

HDMI cable assemblies can fall into one of two categories: Category 1, supporting TMDS clock frequencies up to 74.25MHz and Category 2, supporting TMDS clock frequencies up to 340MHz.

HDMI cable assemblies are specified and tested using two different eye measurement procedures: “non-equalized” (without application of the Reference Cable Equalizer) and “equalized” (with application of the Reference Cable Equalizer).

- Non-equalized eye diagram specification – when driven by any TMDS input waveform meeting the Source eye diagram mask requirements of Figure 4-18 at the tested TMDS clock frequency, the HDMI cable assembly shall produce a TMDS output waveform that meets the Sink eye diagram mask of Figure 4-20.
- Equalized eye diagram specification – when driven by any TMDS input waveform meeting the Source eye diagram mask requirements of Figure 4-18 at the tested TMDS clock frequency, the TMDS output waveform of the cable shall meet the post-equalized eye diagram mask of Figure 4-20 after application of the Reference Cable Equalizer.

The application of these two different specifications depends upon the cables frequency rating:

- Category 1 (74.25MHz): The cable shall meet either:
  A) the parameters specified for Category 1 cables in Table 4-21, or,
  B) the non-equalized eye diagram requirements at 74.25MHz.
- Category 2 (>74.25MHz): The cable shall meet either
  A) the parameters specified for Category 2 cables in Table 4-21, or,
  B) all of:
    - the non-equalized eye diagram requirements at 165MHz and,
    - the equalized eye diagram requirements at 340MHz and,
    - the attenuation limits specified in Figure 4-24 if the cable is a passive copper cable or in Figure 4-25 if the cable has equalization circuitry embedded within.

Table 4-21 Cable Assembly TMDS Parameters

Parameter	Category 1 (74.25MHz)	Category 2 (>74.25MHz)
Maximum Cable Assembly Intra-Pair Skew	151psec	111psec
Maximum Cable Assembly Inter-Pair Skew	2.42nsec	1.78nsec
Far-end Crosstalk	< -26dB	< -26dB
Attenuation	See Figure 4-22	See Figure 4-23
300kHz - 825MHz	< 8dB	< 5dB
825MHz - 2.475GHz	< 21dB	< 5dB...< 12dB
2.475GHz - 4.125GHz	< 30dB	< 12dB...< 20dB
4.125GHz – 5.1GHz	- -	< 20dB...< 25dB
Differential Impedance		
Connection point and transition area: Up to 1nsec**	100 ohms ±15%	
Cable area: 1nsec – 2.5nsec:**	100 ohms ±10%	

*** Measurement point for TDR measurement of impedance.*

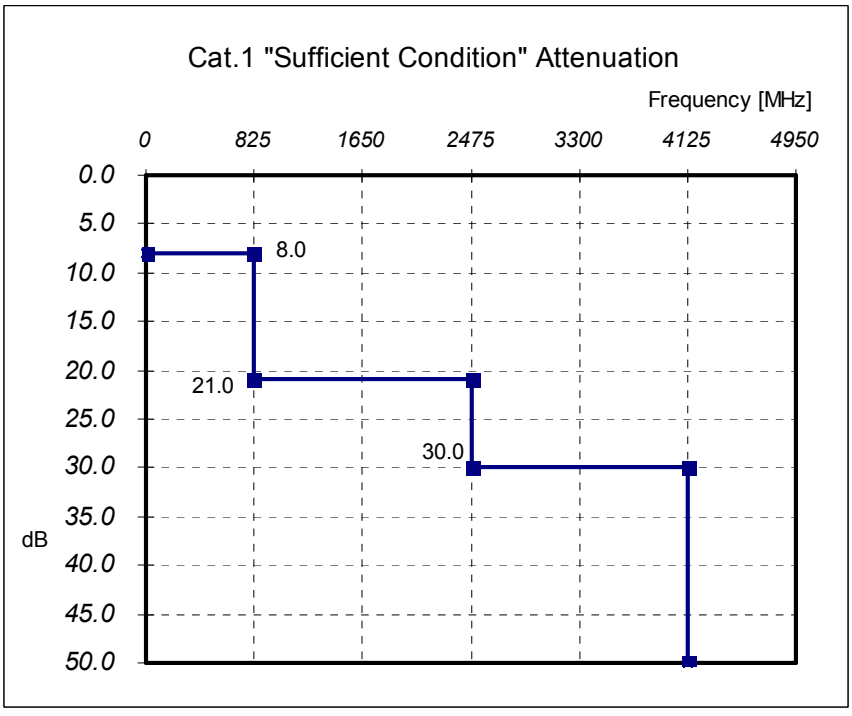


Figure 4-22 Category 1 Cable Attenuation Limits – Sufficient Condition

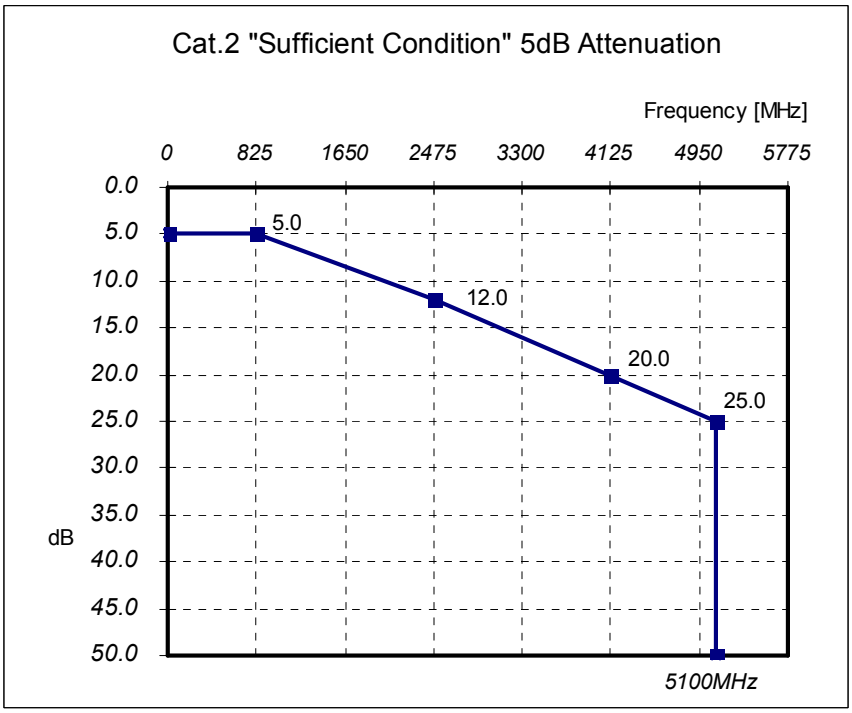


Figure 4-23 Category 2 Cable Attenuation Limits – Sufficient Condition

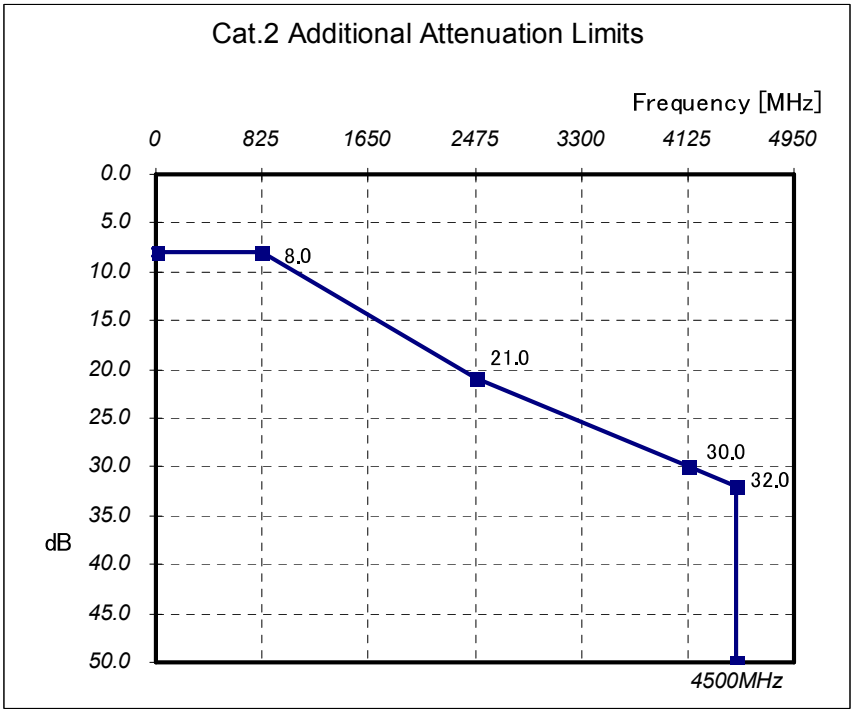


Figure 4-24 Category 2 Additional Attenuation Limits For Copper Cable

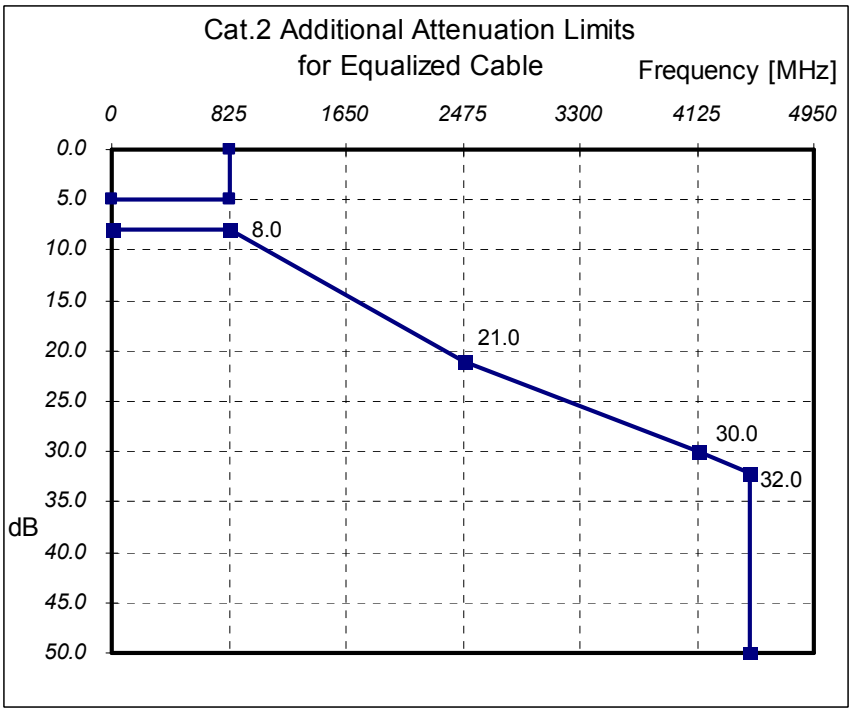


Figure 4-25 Category 2 Additional Attenuation Limits for Equalized Cable

4.2.7 +5V Power Signal

The HDMI connector provides a pin allowing the Source to supply +5.0 Volts to the cable and Sink.

All HDMI Sources shall assert the +5V Power signal whenever the Source is using the DDC or TMDS signals. The voltage driven by the Source shall be within the limits specified for TP1 voltage in Table 4-22. An HDMI Source shall have +5V Power signal over-current protection of no more than 0.5A.

All HDMI Sources shall be able to supply a minimum of 55 mA to the +5V Power pin.

A Sink shall not draw more than 50 mA of current from the +5V Power pin. When the Sink is powered on, it can draw no more than 10mA of current from the +5V Power signal. A Sink shall assume that any voltage within the range specified for TP2 voltage in Table 4-22 indicates that a Source is connected and applying power to the +5V Power signal.

A Cable Assembly shall be able to supply a minimum of 50mA to the +5V Power pin to a Sink, even when connected to a Source supplying no more than 55mA.

The return for the +5V Power signal is DDC/CEC Ground signal.

Table 4-22 +5V Power Pin Voltage

Item	Min	Max
TP1 voltage	4.8 Volts	5.3 Volts
TP2 voltage	4.7 Volts	5.3 Volts

4.2.8 DDC

The Display Data Channel (DDC) I/Os and wires (SDA, SCL, DDC/CEC Ground), shall meet the requirements specified in the I^2C-bus Specification, version 2.1, Section 15 for "Standard-Mode" devices. Note that the discussions of high capacitance environments in the I^2C-bus Specification, section 17.2, "Switched pull-up circuit for Fast-mode I^2C-bus", may be applied to the HDMI environment as well.

HDMI devices shall have DDC electrical characteristics complying with the values shown in Table 4-23 and Table 4-24.

The exact method and measurement procedure is written in HDMI Compliance Test Specification. In some cases, buffers or I^2C "accelerators", may be inserted in the cable as long as all I^2C timing requirements are met.

Table 4-23 Maximum Capacitance of DDC line

Item	HDMI Source	Cable Assembly	HDMI Sink
SDA – DDC/CEC Ground	50pF	700pF	50pF
SCL – DDC/CEC Ground	50pF	700pF	50pF

Table 4-24 Pull-up Resistance on DDC Lines

Item	Value
Source Pull-up resistors for SCL and SDA signals	minimum 1.5k ohms, maximum 2.0k ohms
Sink Pull-up resistors for SCL signal	47k ohms, ±10%

4.2.9 Hot Plug Detect Signal (HPD)

The ground reference for the Hot Plug Detect signal is the DDC/CEC Ground pin.

Table 4-25 Required Output Characteristics of Hot Plug Detect Signal

Item	Value
High voltage level (Sink)	Minimum 2.4 Volts, Maximum 5.3 Volts
Low voltage level (Sink)	Minimum 0 Volts, Maximum 0.4 Volts
Output resistance	1000 ohms ±20%

Table 4-26 Required Detection Levels for Hot Plug Detect Signal

Item	Value
High voltage level (Source)	Minimum 2.0 Volts, Maximum 5.3 Volts
Low voltage level (Source)	Minimum 0 Volts, Maximum 0.8 Volts

Note that many Sink devices simply connect the HPD signal to the +5V Power signal through a 1000 ohm resistor. It may therefore be necessary for a Source to pull-down the HPD signal in order to reliably differentiate between a floating (disconnected) HPD and a high voltage level HPD signal.

4.2.10 CEC Line

The following line characteristics are required for all products, including those that do not implement the CEC protocol. Further requirements for those devices that implement the CEC protocol are given in Supplement 1. The ground reference for the CEC signal is the DDC/CEC Ground signal.

Table 4-27 CEC line Electrical Specifications for all Configurations

Item	Rule / Description	Value
Line connectivity	CEC lines from all HDMI inputs (if present) and a single HDMI output (if present) shall be interconnected. However, the following exceptions are allowed: A device that has no HDMI output is allowed to have separate CEC lines for each HDMI connector if that device implements CEC protocol and takes a logical address of 0 on each CEC line. Due to the complexity of handling multiple active CEC lines, this is discouraged. A device (typically a TV or media receiver box) that is acting as the CEC root device shall not connect the CEC line to any HDMI output. Maximum resistance of interconnected CEC line between any two HDMI connectors:	5Ω
Power-off characteristics	A device with power removed shall not degrade communication between other CEC devices (e.g. the line shall not be pulled down by the powered off device). Maximum CEC line leakage current in off (unpowered) state	1.8μA
CEC Line Capacitance	Maximum capacitance load of a device (excluding cable)	200pF
	Maximum capacitance load of a Cable Assembly	700pF

4.2.11 Robustness Requirements

No damage to the HDMI Source or Sink can result from the shorting of any combination of signals on any connector. If two HDMI Sources are connected together with a single cable, no damage can occur to either of the Sources. If two HDMI Sinks are connected together with a single cable, no damage can occur to either of the Sinks.

5 Signaling And Encoding

5.1 Overview

5.1.1 Link Architecture

As shown in Figure 5-1, an HDMI link includes three TMDS Data channels and a single TMDS Clock channel. The TMDS Clock channel constantly runs at a rate proportional to the pixel rate of the transmitted video. During every cycle of the TMDS Clock channel, each of the three TMDS data channels transmits a 10-bit character. This 10-bit word is encoded using one of several different coding techniques.

The input stream to the Source's encoding logic will contain video pixel, packet and control data. The packet data consists of audio and auxiliary data and associated error correction codes.

These data items are processed in a variety of ways and are presented to the TMDS encoder as either 2 bits of control data, 4 bits of packet data or 8 bits of video data per TMDS channel. The Source encodes one of these data types or encodes a Guard Band character on any given clock cycle.

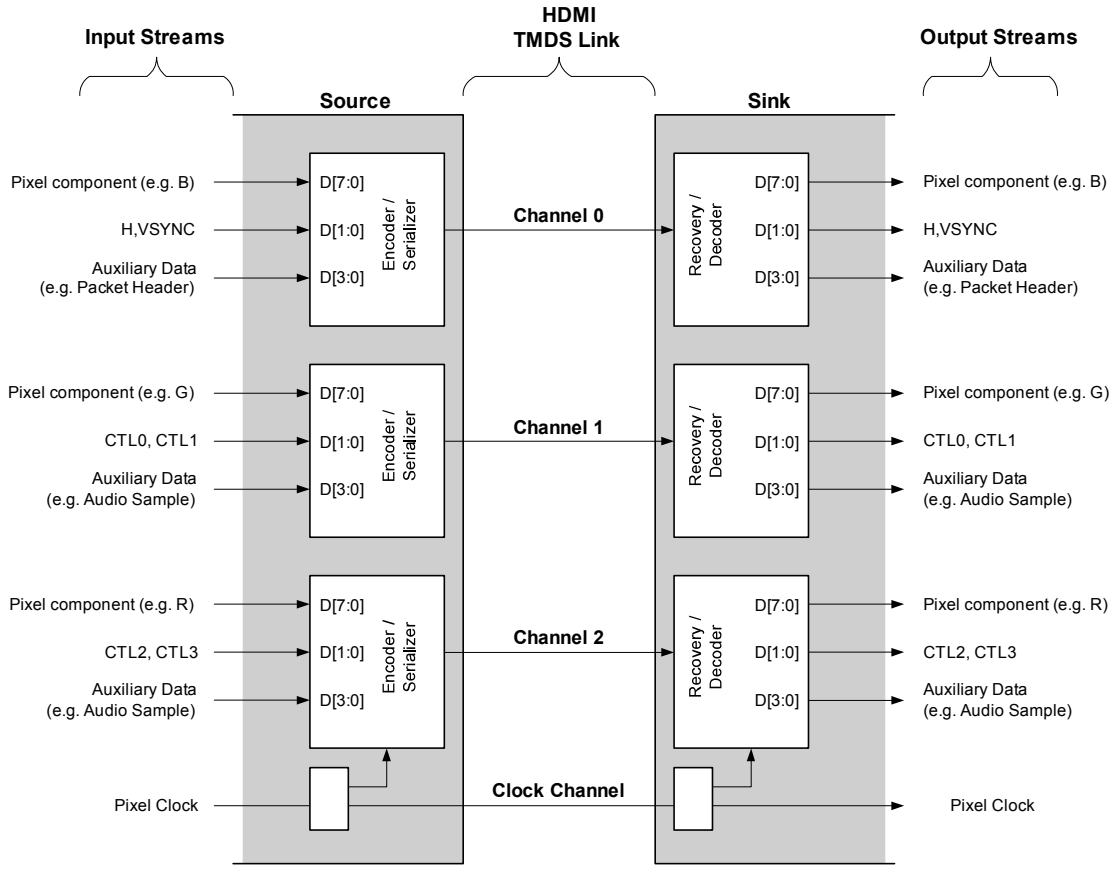


Figure 5-1 HDMI Encoder/Decoder Overview

5.1.2 Operating Modes Overview

The HDMI link operates in one of three modes: Video Data Period, Data Island period, and Control period. During the Video Data Period, the active pixels of an active video line are transmitted. During the Data Island period, audio and auxiliary data are transmitted using a series of packets. The Control period is used when no video, audio, or auxiliary data needs to be transmitted. A Control Period is required between any two periods that are not Control Periods.

An example of each period placement is shown in the following figure.

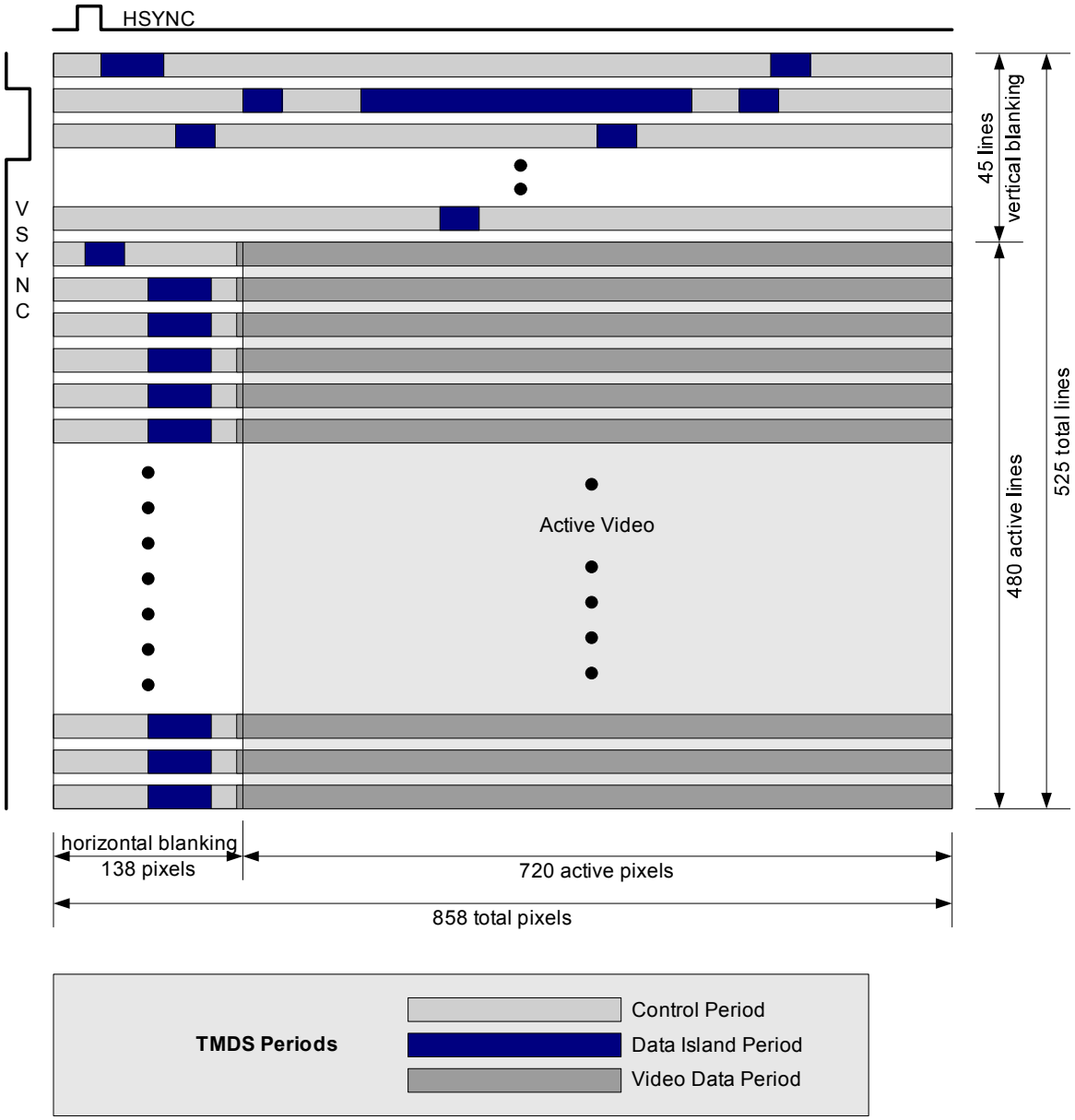


Figure 5-2 Informative Example: TMDS periods in 720x480p video frame

Video Data Periods use transition minimized coding to encode 8 bits per channel, or 24 bits total per pixel.

Data Island Periods are encoded using a similar transition minimized coding, TMDS Error Reduction Coding (TERC4), which transmits 4 bits per channel, or 12 bits total per TMDS clock period.

During Control Periods, 2 bits per channel, or 6 bits total are encoded per TMDS clock using a transition maximized encoding. These 6 bits are HSYNC, VSYNC, CTL0, CTL1, CTL2 and CTL3. Near the end of every Control Period, a Preamble, using the CTLx bits, indicates whether the next Data Period is a Video Data Period or a Data Island Period.

Each Video Data Period and Data Island Period starts with a Leading Guard Band designed to provide robust determination of the transition from the Control Period to the Data Period. This Leading Guard Band consists of two special characters.

The Data Island Period is also protected by a Trailing Guard Band, which is designed to provide a robust determination of the transition to Control Period.

The following table shows Encoding type used and data transmitted during each operating mode.

Table 5-1 Encoding Type and Data Transmitted

Period	Data Transmitted	Encoding Type
Video Data	Video Pixels	Video Data Coding (8 bits converted to 10 bits)
	(Guard Band)	(Fixed 10 bit pattern)
Data Island	Packet Data - Audio Samples - InfoFrames HSYNC, VSYNC	TERC4 Coding (4 bits converted to 10 bits)
	(Guard Band)	(Fixed 10 bit pattern)
Control	Control - Preamble - HSYNC, VSYNC	Control Period Coding (2 bits converted to 10 bits)

5.2 Operating Modes

5.2.1 Control Period

Control Period is used for transmission of the Preamble. The Control Period is also used by the Sink for character synchronization.

The HDCP-specified Enhanced Encryption Status Signaling ENC_EN code (CTL0:3=1001) shall not be used except as a correct ENC_EN during the HDCP-specified window of opportunity.

5.2.1.1 Preamble

Immediately preceding each Video Data Period or Data Island Period is the Preamble. This is a sequence of eight identical Control characters that indicate whether the upcoming data period is a Video Data Period or is a Data Island. The values of CTL0, CTL1, CTL2, and CTL3 indicate the type of data period that follows. The remaining Control signals, HSYNC and VSYNC, may vary during this sequence.

There are only two legal Preamble characters:

Table 5-2 Preambles for Each Data Period Type

CTL0	CTL1	CTL2	CTL3	Data Period Type
1	0	0	0	Video Data Period
1	0	1	0	Data Island Period

The Video Data Period type indicates that the following data period contains video data, beginning with a Video Guard Band.

The Data Island type indicates that the following data period is an HDMI compliant Data Island, beginning with a Data Island Guard Band.

The transition from TMDS control characters to Guard Band characters following this sequence identifies the start of the Data Period.

The Data Island Preamble control code (CTL0:3=1010) shall not be transmitted except for correct use during a Preamble period.

5.2.1.2 Character Synchronization

The TMDS Sink needs to determine the location of character boundaries in the serial data streams. Once character boundaries are established on all data channels, the Sink is defined to be synchronized to the serial streams, and may recover TMDS characters from the data channels for decode. The TMDS data stream provides periodic cues for decoder synchronization.

The TMDS characters used during the Video Data Period and Data Island Period contain five or fewer transitions, while the TMDS characters used during the Control Period contain seven or more transitions. The high-transition content of the characters transmitted during the Control Period form the basis for character boundary synchronization at the decoder. While these characters are not individually unique in the serial data stream, they are sufficiently alike that the decoder may uniquely detect the presence of a succession of them during transmitted synchronization intervals. The exact algorithm for this detection is an implementation detail beyond the scope of this document, but minimum conditions for Sink synchronization are defined.

The Sink is required to establish synchronization with the data stream during any Control Period greater than or equal to $t_{S,min}$ (12) characters in length.

The Source is also required to occasionally transmit an Extended Control Period per Table 5-4.

Table 5-3 TMDS Link Timing Parameters

Parameter	Description	Value	Unit
$t_{S,min}$	Minimum duration Control Period	12	T_{PIXEL}

Table 5-4 Extended Control Period Parameters

Parameter	Description	Value	Unit
t_{EXTS,max_delay}	Maximum time between Extended Control Periods	50	msec
$t_{EXTS,min}$	Minimum duration Extended Control Period	32	T_{PIXEL}

5.2.2 Video Data Period

Video data periods are used to carry the pixels of an active video line.

Each Video Data Period is preceded by a Preamble, described above.

Following the Preamble, the Video Data Period begins with a two character Video Leading Guard Band. There is no Trailing Guard Band for the Video Data Period.

During active video periods, 24 bits of pixel data are encoded using TMDS transition minimized encoding during each TMDS clock period.

5.2.2.1 Video Guard Band

Table 5-5 Video Leading Guard Band Values

```
case (TMDS Channel Number):
       0:  q_out[9:0] = 0b1011001100;
       1:  q_out[9:0] = 0b0100110011;
       2:  q_out[9:0] = 0b1011001100;
endcase
```

5.2.3 Data Island Period

5.2.3.1 Data Island Overview

Data Islands are used to carry packets of audio sample data and auxiliary data. This auxiliary data includes InfoFrames and other data describing the active audio or video stream or describing the Source.

Each Data Island is preceded by a Preamble, described above.

Following the Preamble, each Island starts with a Leading Guard Band. The first packet of the Data Island then follows.

During every TMDS clock period of the Data Island, including the Guard Band, bits 0 and 1 of TMDS Channel 0 transmit an encoded form of HSYNC and VSYNC.

Bit 2 of TMDS Channel 0 is used to transmit the Packet Header. All four bits of TMDS Channels 1 and 2 are used for the Packet data as shown in Figure 5-3. Each packet is 32 pixels long and is protected by BCH ECC for error correction and detection purposes.

During the Data Island, each of the three TMDS channels transmits a series of 10-bit characters encoded from a 4-bit input word, using TMDS Error Reduction Coding (TERC4). TERC4 significantly reduces the error rate on the link by choosing only 10-bit codes with high inherent error avoidance.

The last two characters of the Data Island, following the last packet, is the Trailing Guard Band.

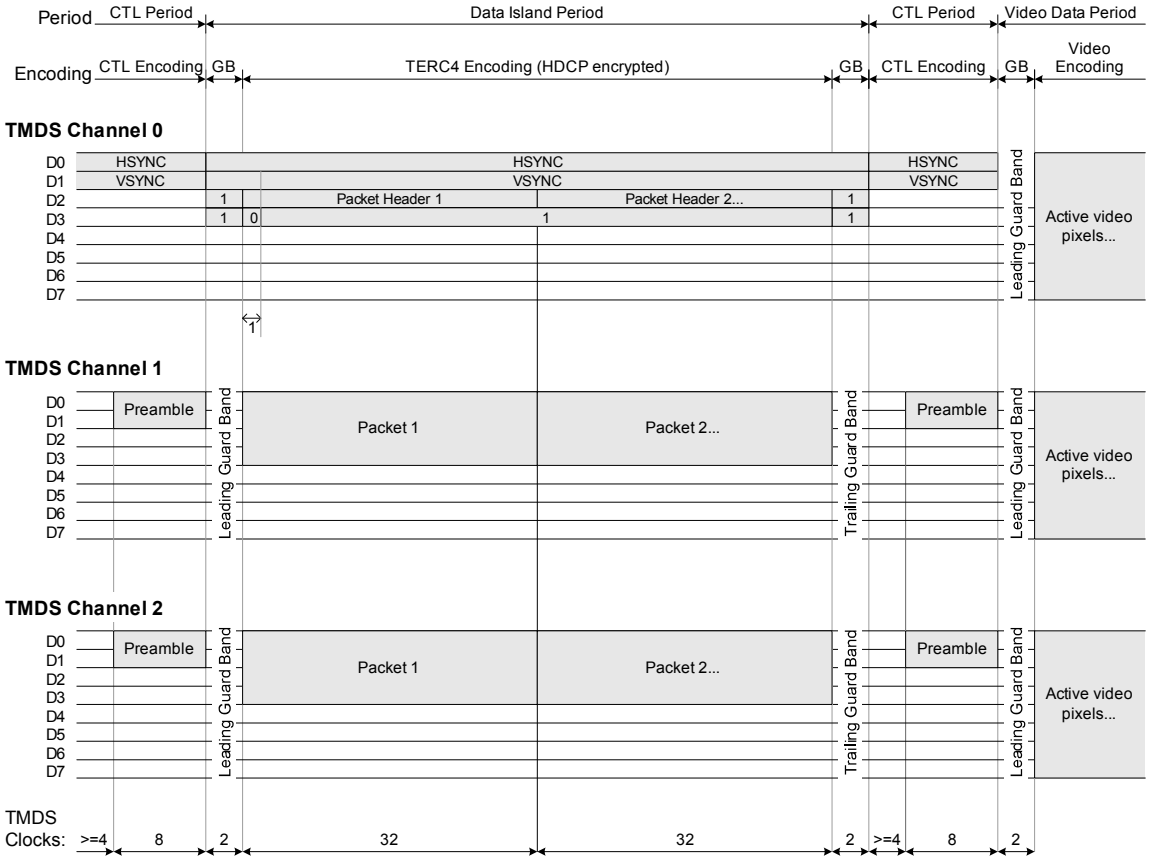


Figure 5-3 TMDS Periods and Encoding

Following the Data Island, all three channels revert to transmitting control characters.

5.2.3.2 Island Placement and Duration

The Source is required to determine the temporal placement and duration of the Data Island with respect to the video signal's horizontal and vertical blanking periods and synchronization signals. It shall do so following the rules stated below.

All TMDS Control Periods shall be at least $t_{S,min}$ (12) characters (pixels) long.

The Data Island shall contain at least one packet, limiting its minimum size to 36 pixels.

Islands shall contain an integer number of packets. In order to assure the reliability of the data within the Data Island, they shall be limited to 18 packets or fewer.

Zero, one or more Data Islands can occur between subsequent video data periods.

While transmitting video, at least one Data Island shall be transmitted during every two video fields.

5.2.3.3 Data Island Guard Bands

The first two data characters within the Data Island are the Leading Guard Band. The last two data characters within the Data Island are the Trailing Guard Band.

During the Data Island Guard Bands, Channel 0 is encoded as one of four TERC4 values. These TERC4 values (D[3:0]) are 0xC, 0xD, 0xE and 0xF, depending upon the values of HSYNC and VSYNC.

Table 5-6 Data Island Leading and Trailing Guard Band Values

```
case (TMDS Channel Number):
       0:  q_out[9:0] = n.a.;
       1:  q_out[9:0] = 0b0100110011;
       2:  q_out[9:0] = 0b0100110011;
endcase
```

5.2.3.4 Data Island Packet Construction

All data within a Data Island is contained within 32 clock Packets. Packets consist of a Packet Header, a Packet Body (consisting of four Subpackets), and associated error correction bits. Each Subpacket includes 56 bits of data and is protected by an additional 8 bits of BCH ECC parity bits.

Subpacket 0 plus its corresponding parity bits make up BCH Block 0. This block is mapped onto bit 0 of both Channel 1 and Channel 2. In this way, the 64 bits of BCH Block 0 are transferred over the course of 32 pixels. Likewise, BCH Block 1 (Subpacket 1 plus parity) is mapped onto bit 1 of both Channels 1 and 2.

In the tables below, Header bytes are indicated as HB0, HB1, and HB2 and Subpacket bytes are indicated as SB0 to SB6.

Subpacket 0 bytes 0 through 6 (SB0-SB6) are also designated Packet bytes 0 to 6 (PB0-PB6).

Subpacket 1 bytes 0 through 6 (SB0-SB6) are also designated Packet bytes 7 to 13 (PB7-PB13).

Subpacket 2 bytes 0 through 6 (SB0-SB6) are also designated Packet bytes 14 to 20 (PB14-PB20).

Subpacket 3 bytes 0 through 6 (SB0-SB6) are also designated Packet bytes 21 to 27 (PB21-PB27).

This is illustrated in Figure 5-4.

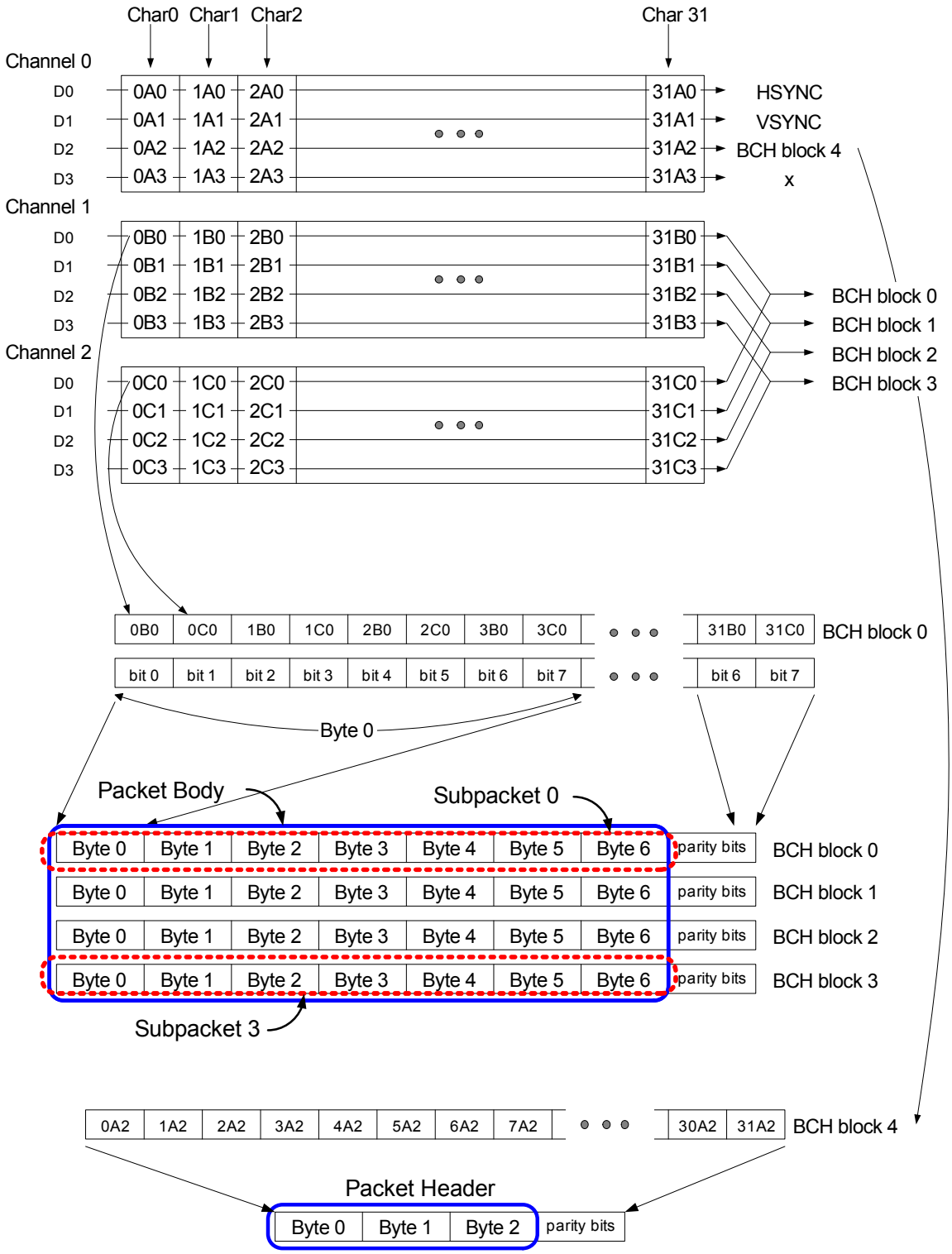


Figure 5-4 Data Island Packet and ECC Structure

5.2.3.5 Data Island Error Correction

To improve the reliability of the data and to improve the detection of bad data, Error Correction Code (ECC) parity is added to each packet. BCH(64,56) and BCH(32,24) are generated by the polynomial G(x) shown in Figure 5-5.

$G(x)=1+x^6+x^7+x^8$ (127 count repetition cycle).

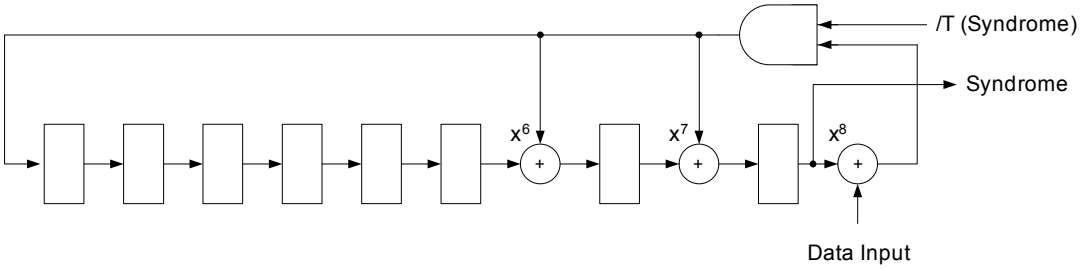


Figure 5-5 Error Correction Code generator

5.3 Data Island Packet Definitions

5.3.1 Packet Header

Packet Headers contain 24 data bits with an additional 8 bits of BCH(32,24) ECC parity. These parity bits are calculated over the 24 bits of the Packet Header.

A Packet Header includes an 8-bit Packet Type and 16 bits of packet-specific data.

A Sink shall be able to receive, with no adverse effects, any packet defined in the HDMI 1.0 specification including any InfoFrame Packet with an InfoFrame Type defined in CEA-861-D.

Table 5-7 Packet Header

Byte \ Bit #	7	6	5	4	3	2	1	0
HB0	Packet Type							
HB1	packet-specific data							
HB2	packet-specific data							

Table 5-8 shows the available packet types.

Table 5-8 Packet Types

Packet Type Value		Packet Type	Described in Section
0x00		Null	5.3.2
0x01		Audio Clock Regeneration (N/CTS)	5.3.3
0x02		Audio Sample (L-PCM and IEC 61937 compressed formats)	5.3.4
0x03		General Control	5.3.6
0x04		ACP Packet	5.3.7
0x05		ISRC1 Packet	5.3.8
0x06		ISRC2 Packet	"
0x07		One Bit Audio Sample Packet	5.3.9
0x08		DST Audio Packet	5.3.10
0x09		High Bitrate (HBR) Audio Stream Packet (IEC 61937)	5.3.11
0x0A		Gamut Metadata Packet	5.3.12
0x80+InfoFrame Type		InfoFrame Packet	5.3.5
	0x81	Vendor-Specific InfoFrame	- -
	0x82	AVI InfoFrame*	8.2.1
	0x83	Source Product Descriptor InfoFrame	- -
	0x84	Audio InfoFrame*	8.2.2
	0x85	MPEG Source InfoFrame	- -

** See Section 8.2 for the packet layout for these InfoFrames*

5.3.2 Null Packet

Null packets can be used by the Source anytime. All bytes of a Null packet are undefined and shall contain only zero values. An HDMI Sink shall ignore bytes HB1 and HB2 of the Null Packet Header and all bytes of the Null Packet Body.

Table 5-9 Null Packet Header

Byte \ Bit #	7	6	5	4	3	2	1	0
HB0	0	0	0	0	0	0	0	0
HB1	0	0	0	0	0	0	0	0
HB2	0	0	0	0	0	0	0	0

5.3.3 Audio Clock Regeneration Packet

Audio Clock Regeneration Packets contain both the N and CTS values used in the Audio Clock Regeneration process. The four Subpackets each contain the same Audio Clock Regeneration Subpacket. An HDMI Sink shall ignore bytes HB1 and HB2 of the Audio Clock Regeneration Packet header.

Table 5-10 Audio Clock Regeneration Packet Header

Byte \ Bit #	7	6	5	4	3	2	1	0
HB0	0	0	0	0	0	0	0	1
HB1	0	0	0	0	0	0	0	0
HB2	0	0	0	0	0	0	0	0

Table 5-11 Audio Clock Regeneration Subpacket

Byte \ Bit #	7	6	5	4	3	2	1	0
SB0	0	0	0	0	0	0	0	0
SB1	0	0	0	0	CTS.19	-	-	CTS.16
SB2	CTS.15	-	-	-	-	-	-	CTS.8
SB3	CTS.7	-	-	-	-	-	-	CTS.0
SB4	0	0	0	0	N.19	-	-	N.16
SB5	N.15	-	-	-	-	-	-	N.8
SB6	N.7	-	-	-	-	-	-	N.0

- N [20 bits] value of audio clock regeneration “N”
- CTS [20 bits] Cycle Time Stamp

CTS values of zero are used to indicate no new value of CTS.

5.3.4 Audio Sample Packet

L-PCM and some IEC 61937 compressed audio formats are carried using Audio Sample Packets. Audio Sample Packets consist of one to four Audio Samples. These may be different samples or different partial samples (i.e. 2 of 6 channels). The configuration of the Subpackets is determined by the layout and sample_present bits in the header. This is described in detail in Section 7.6, *Audio Data Packetization*.

Table 5-12 Audio Sample Packet Header

Byte \ Bit #	7	6	5	4	3	2	1	0
HB0	0	0	0	0	0	0	1	0
HB1	0	0	0	layout	sample_ present.sp3	sample_ present.sp2	sample_ present.sp1	sample_ present.sp0
HB2	B.3	B.2	B.1	B.0	sample_flat .sp3	sample_flat .sp2	sample_flat .sp1	sample_flat .sp0

- layout: [1 bit] indicates which of two possible Subpacket/audio sample layouts are used. See Section 7.6, *Audio Data Packetization*.
- sample_present.spX [4 fields, 1 bit each] indicates if Subpacket X contains audio sample(s).
- sample_flat.spX [4 fields, 1 bit each] indicates if Subpacket X represents a "flatline" sample. Only valid if "sample_present.spX" is set.
- B.X [4 fields, 1 bit each] B.X =1 if Subpacket X contains the first frame in a 192 frame IEC 60958 Channel Status block; B.X = 0 otherwise

Table 5-13 Audio Sample Subpacket

Byte \ Bit #	7	6	5	4	3	2	1	0
SB0	L.11	-	-	-	-	-	-	L.4
SB1	L.19	-	-	-	-	-	-	L.12
SB2	L.27	-	-	-	-	-	-	L.20
SB3	R.11	-	-	-	-	-	-	R.4
SB4	R.19	-	-	-	-	-	-	R.12
SB5	R.27	-	-	-	-	-	-	R.20
SB6	P_R	C_R	U_R	V_R	P_L	C_L	U_L	V_L

- L.X: [24 fields, 1 bit each] Bit corresponding to Time Slot X from first ("left") sub-frame per IEC 60958-1, page 15
- R.X: [24 fields, 1 bit each] Bit corresponding to Time Slot X from second ("right") sub-frame per IEC 60958-1, page 15
- V_L: [1 bit] Valid bit from first sub-frame
- V_R: [1 bit] Valid bit from second sub-frame
- U_L: [1 bit] User Data bit from first sub-frame
- U_R: [1 bit] User Data bit from second sub-frame
- C_L: [1 bit] Channel Status bit from first sub-frame
- C_R: [1 bit] Channel Status bit from second sub-frame
- P_L: [1 bit] Parity bit from first sub-frame (even parity)

DELL/HP/LENOVO EXHIBIT 1015
Page 438

- P_R: [1 bit] Parity bit from second sub-frame (even parity)

5.3.5 InfoFrame Packet

All InfoFrames defined in CEA-861-D may be carried across HDMI using the HDMI InfoFrame packet. InfoFrames not defined in CEA-861-D or in this specification shall not be transmitted.

Each HDMI InfoFrame Packet carries a single CEA InfoFrame, as shown below[2].[pw99] Note that HDMI places additional requirements on several InfoFrames that are not covered by CEA-861-D. For these additional details and restrictions, see Section 8.2.

Table 5-14 InfoFrame Packet Header

Byte \ Bit #	7	6	5	4	3	2	1	0
HB0	1	InfoFrame Type						
HB1	InfoFrame_version							
HB2	0	0	0	InfoFrame_length				

- InfoFrame Type [7 bits] least significant 7 bits of the InfoFrame type code as per CEA-861-D.
- InfoFrame_version [1 byte] version number of InfoFrame as per CEA-861-D.
- InfoFrame_length [5 bits] InfoFrame length in bytes as per CEA-861-D. This length does not include any of the bytes in the Packet Header nor the checksum byte. The maximum value for this field is 27 (0x1B).

Table 5-15 InfoFrame Packet Contents

Byte \ Bit #	7	6	5	4	3	2	1	0
PB0	Checksum							
PB1	Data Byte 1							
PB2	Data Byte 2							
PB3...PB26	-							
PB27	Data Byte 27							

- Checksum [1 byte] Checksum of the InfoFrame. The checksum shall be calculated such that a byte-wide sum of all three bytes of the Packet Header and all valid bytes of the InfoFrame Packet contents (determined by InfoFrame_length), plus the checksum itself, equals zero.

[2] An earlier version of CEA-861-D, CEA-861B, had a method for encapsulating multiple CEA InfoFrames into a single CEA InfoPacket. HDMI has its own packet structure and therefore CEA Info*Packets* are not used.

- Data Byte x [27 fields, 1 byte each] Data Byte X of the InfoFrame as defined in CEA-861-D. See [HDMI Specification] Section 8.2 for more information.

5.3.6 General Control Packet

The General Control packet header contains no data. Bytes HB1 and HB2 shall be ignored by the Sink. The General Control packet body shall contain four identical subpackets, defined in Table 5-17, below.

Table 5-16 General Control Packet Header

Byte \ Bit #	7	6	5	4	3	2	1	0
HB0	0	0	0	0	0	0	1	1
HB1	0	0	0	0	0	0	0	0
HB2	0	0	0	0	0	0	0	0

Table 5-17 General Control Subpacket

Byte \ Bit #	7	6	5	4	3	2	1	0
SB0	0	0	0	Clear_AVMUTE	0	0	0	Set_AVMUTE
SB1	PP3	PP2	PP1	PP0	CD3	CD2	CD1	CD0
SB2	0	0	0	0	0	0	0	Default_Phase
SB3	0	0	0	0	0	0	0	0
SB4	0	0	0	0	0	0	0	0
SB5	0	0	0	0	0	0	0	0
SB6	0	0	0	0	0	0	0	0

- Set_AVMUTE [1 bit] Set the AVMUTE flag. (See description below).
- Clear_AVMUTE [1bit] Clear the AVMUTE flag. (See description below).
- PP [4 bits] Pixel Packing Phase. (See description in section 6.5.3.)
- CD [4 bits] Color Depth. (See description in section 6.5.3.)
- Default_Phase [1 bit] Default Phase. (See description in section 6.5.3.)

The General Control Packet contains fields for indicating AVMUTE information and color-depth information. Each transmitted GCP may contain valid indications for AVMUTE and/or color-depth or may contain no information (all fields zero).

General Control packets indicating Set_AVMUTE or Clear_AVMUTE may only be transmitted between the active edge of VSYNC and 384 pixels following this edge. A Source may not send a General Control Packet with the Clear_AVMUTE and Set_AVMUTE flags set simultaneously.

Source transmission of the General Control Packet is optional. Sinks may optionally interpret General Control Packet contents. Sinks shall be capable of receiving any General Control Packet.

The General Control packet's Set_AVMUTE and Clear_AVMUTE flags may be used by a Source to reduce the negative impact on the Sink of TMDS clock changes or interruptions. Use of the AVMUTE function may prevent spurious pops or noises in the audio during these clock changes.

When AVMUTE is set, the Sink may assume that no valid audio or video data is being received. The Sink may optionally apply a mute function to the audio data and/or a blank function to the video.

5.3.7 Audio Content Protection Packet (ACP)

A Source may use the ACP Packet to convey content-related information regarding the active audio stream.

See Section 9.3 for rules regarding the use of the ACP packet.

The following tables show the packetization of the ACP Packet.

Table 5-18 ACP Packet Header

Byte \ Bit #	7	6	5	4	3	2	1	0
HB0	Packet Type = 0x04							
HB1	ACP_Type							
HB2	Reserved (0)							

- ACP_Type [1 byte] Content protection type (see Section 9.3 for usage):
  0x00 = Generic Audio
  0x01 = IEC 60958-Identified Audio
  0x02 = DVD-Audio
  0x03 = Super Audio CD
  0x04...0xFF Reserved

Table 5-19 ACP Packet contents

Packet Byte #	7	6	5	4	3	2	1	0
PB0-PB27	ACP_Type_Dependent (Dependent upon ACP_Type value)							

- ACP_Type_Dependent [28 bytes] Contents are dependent upon ACP_Type field. See Section 9.3 for usage.

5.3.8 ISRC Packets

A Source may use the ISRC packets to transmit a UPC/EAN or ISRC code. See Section 8.8 for rules regarding the use of the ISRC packets.

Table 5-20 ISRC1 Packet Header

Byte \ Bit #	7	6	5	4	3	2	1	0
HB0	Packet Type = 0x05							
HB1	ISRC_ Cont	ISRC_ Valid	Reserved (0)			ISRC_Status		
HB2	Reserved (0)							

- ISRC_Cont [1 bit] ISRC Continued (in next packet). See Section 8.8 for usage.
- ISRC_Status [3 bits] See Section 8.8 for usage.
- ISRC Valid [1 bit]: This bit is set only when data located in ISRC_Status field and UPC_EAN_ISRC_xx field are valid. When Source cannot obtain complete data for these fields, ISRC_Valid may be 0.

Table 5-21 ISRC1 Packet contents

Packet Byte #	7	6	5	4	3	2	1	0
PB0	UPC_EAN_ISRC_0							
PB1	UPC_EAN_ISRC_1							
PB2	UPC_EAN_ISRC_2							
PB3	UPC_EAN_ISRC_3							
PB4	UPC_EAN_ISRC_4							
PB5	UPC_EAN_ISRC_5							
PB6	UPC_EAN_ISRC_6							
PB7	UPC_EAN_ISRC_7							
PB8	UPC_EAN_ISRC_8							
PB9	UPC_EAN_ISRC_9							
PB10	UPC_EAN_ISRC_10							
PB11	UPC_EAN_ISRC_11							
PB12	UPC_EAN_ISRC_12							
PB13	UPC_EAN_ISRC_13							
PB14	UPC_EAN_ISRC_14							
PB15	UPC_EAN_ISRC_15							
PB16-PB27	Reserved (0)							

- UPC_EAN_ISRC_xx [16 fields, 1 byte each] UPC/EAN or ISRC byte xx. See Section 8.8 for usage.

Bytes PB16-PB27 shall be set to a value of 0.

Table 5-22 ISRC2 Packet Header

Byte \ Bit #	7	6	5	4	3	2	1	0
HB0	Packet Type = 0x06							
HB1	Reserved (0)							
HB2	Reserved (0)							

Table 5-23 ISRC2 Packet contents

Packet Byte #	7	6	5	4	3	2	1	0
PB0	UPC_EAN_ISRC_16							
PB1	UPC_EAN_ISRC_17							
PB2	UPC_EAN_ISRC_18							
PB3	UPC_EAN_ISRC_19							
PB4	UPC_EAN_ISRC_20							
PB5	UPC_EAN_ISRC_21							
PB6	UPC_EAN_ISRC_22							
PB7	UPC_EAN_ISRC_23							
PB8	UPC_EAN_ISRC_24							
PB9	UPC_EAN_ISRC_25							
PB10	UPC_EAN_ISRC_26							
PB11	UPC_EAN_ISRC_27							
PB12	UPC_EAN_ISRC_28							
PB13	UPC_EAN_ISRC_29							
PB14	UPC_EAN_ISRC_30							
PB15	UPC_EAN_ISRC_31							
PB16-PB27	Reserved (0)							

- UPC_EAN_ISRC_xx [16 fields, 1 byte each] UPC/EAN or ISRC byte xx.

Bytes PB16-PB27 shall be set to a value of 0.

5.3.9 One Bit Audio Sample Packet

One Bit Audio streams are transmitted using the One Bit Audio Sample Packet.

One Bit Audio Packets consist of one to four One Bit Audio Subpackets. These may be different samples or different partial samples (e.g. 2 of 6 channels). The configuration of the Subpackets is determined by the layout and samples_present bits in the header. This is described in detail in Section 7.6, *Audio Data Packetization*.

It is optional for the Source, Sink and Repeater to support the One Bit Audio packet.

Table 5-24 One Bit Audio Packet Header

Byte \ Bit #	7	6	5	4	3	2	1	0
HB0	0	0	0	0	0	1	1	1
HB1	Rsvd (0)	Rsvd (0)	Rsvd (0)	layout	samples_ present.sp3	samples_ present.sp2	samples_ present.sp1	samples_ present.sp0
HB2	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	samples_ invalid.sp3	samples_ invalid.sp2	samples_ invalid.sp1	samples_ invalid.sp0

- layout [1 bit] indicates which of two possible Subpacket/audio sample layouts are used. See Table 5-25 below and Section 7.6, *Audio Data Packetization*.
- samples_present.spX [4 fields, 1 bit each] indicates if Subpacket X contains audio sample data. Samples_present.spX = 1 if subpacket X contains sample data; else = 0.
- samples_invalid.spX [4 fields, 1 bit each] indicates if Subpacket X represents invalid samples. Samples_invalid = 1 if the samples in Subpacket X are invalid; else = 0. This bit is only valid if the relevant "samples_present.spX" is set.

Note that, for One Bit Audio, sample frequency information is carried in the Audio InfoFrame (see section 8.2.2).

Table 5-25 One Bit Audio Subpacket

Byte \ Bit #	7	6	5	4	3	2	1	0
SB0	ChA.7	-	-	-	-	-	-	ChA.0
SB1	ChA.15	-	-	-	-	-	-	ChA.8
SB2	ChA.23	-	-	-	-	-	-	ChA.16
SB3	ChB.7	-	-	-	-	-	-	ChB.0
SB4	ChB.15	-	-	-	-	-	-	ChB.8
SB5	ChB.23	-	-	-	-	-	-	ChB.16
SB6	ChB.27	ChB.26	ChB.25	ChB.24	ChA.27	ChA.26	ChA.25	ChA.24

- ChA.X: [28 fields, 1 bit each] indicates consecutive One Bit Audio samples of the first channel. The most significant bit (ChA.27) is the first sampled bit of the consecutive 28-bit part in the One Bit Audio stream.
- ChB.X: [28 fields, 1 bit each] indicates consecutive One Bit Audio samples of the second channel. The most significant bit (ChB.27) is the first sampled bit of the consecutive 28-bit part in the One Bit Audio stream.

5.3.10 DST Audio Packet

DST (compressed DSD) audio streams are transmitted using the DST Audio Packet.

A DST Audio Packet contains a single DST Audio Packet Body which is filled as audio data becomes available. All identification of channels and other data is embedded in the stream. DST Audio Packet packing is described further in 7.6.3, *DST Packetization*.

It is optional for a Source, Sink or Repeater to support the DST Audio Packet.

Table 5-26 DST Audio Packet Header

Byte \ Bit #	7	6	5	4	3	2	1	0
HB0	0	0	0	0	1	0	0	0
HB1	frame_ start	samples_ invalid	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	DST_normal _double
HB2	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)

- frame_start [1 bit] =1 indicates that this packet is the start of a DST frame; =0 otherwise.
- samples_invalid [1 bit] = 1 if the samples are not valid; = 0 if the samples are valid.
- DST_Normal_Double [1 bit] =0 (“DST_Normal”) indicates that the sample rate equals the transfer rate. =1 (“DST_Double”) indicates that the transfer rate is twice the sample rate. DST_Double rate is used when normal does not have sufficient bandwidth.

Note that, for DST, sample frequency information is carried in the Audio InfoFrame (see section 8.2.2).

Each DST Audio Packet Body contains 224 bits (28 bytes) of DST data. DST stream data is taken in byte order and packed into the DST Audio Packet Body as shown in Table 5-27.

Table 5-27 DST Audio Packet Body

Byte \ Bit #	7	6	5	4	3	2	1	0
PB0	D.7	D.6	D.5	D.4	D.3	D.2	D.1	D.0
PB1	D.15	D.14	D.13	D.12	D.11	D.10	D.9	D.8
-	-	-	-	-	-	-	-	-
PB26	D.215	D.214	D.213	D.212	D.211	D.210	D.209	D.208
PB27	D.223	D.222	D.221	D.220	D.219	D.218	D.217	D.216

- D.X [224 fields, 1 bit each] DST bitstream, beginning with D.0.

5.3.11 High-Bitrate (HBR) Audio Stream Packet

High bitrate (>6.144Mbps) compressed audio streams conforming to IEC 61937 are carried using HBR Audio Stream Packets. Each packet carries four IEC 60958 frames which corresponds to (4x2x16 =) 128 contiguous bits of an IEC 61937 stream. This is described in more detail in Section 7.6.2, *High-Bitrate Audio Stream Packetization*.

Table 5-28 HBR Audio Stream Packet Header

Byte \ Bit #	7	6	5	4	3	2	1	0
HB0	0	0	0	0	1	0	0	1
HB1	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)
HB2	B.3	B.2	B.1	B.0	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)

- B.X [4 fields, 1 bit each] B.X =1 if Subpacket X contains the first frame in a 192 frame IEC 60958 Channel Status block; B.X = 0 otherwise

The HBR Audio Stream Packet uses four subpackets which are identical to the Audio Sample Subpacket shown above in Table 5-13.

5.3.12 Gamut Metadata Packet

Gamut boundary descriptions (GBD) and other gamut-related metadata are carried using the Gamut Metadata Packet. Gamut metadata is further described in Appendix E.

One of several transmission profiles (P0, P1, P2, etc.) can be used when sending GBDs using this packet. The difference between the transmission profiles is primarily the transmission rate,

specifically, the number of packets that may be sent per video field. The transmission profile is indicated in the field GBD_profile.

The lowest transmission profile (P0) also has a specific size limitation; it fits within a single Gamut Metadata Packet. Higher transmission profiles (P1, P2, etc.) can be larger, requiring many packets and possibly ten or more video fields for complete transmission.

Table 5-29 Gamut Boundary Description Metadata Profiles

Transmission Profile Name	EDID Colorimetry Data Block bit	Packets per Video Field	Total Packet Count
P0	MD0	1	1
P1	MD1	1	2-10
P2	MD2	2-10	11-100
P3	*	11-80	101-800

** This bit will be defined in a future specification.*

Gamut metadata may be transmitted that describes the gamut of the currently transmitted video or that of upcoming video. Each time the gamut of the video stream changes in a way that requires transmission of new gamut metadata, a gamut sequence number is incremented. All metatdata packets include two fields, Affected_Gamut_Seq_Num and Current_Gamut_Seq_Num, that together indicate whether the metadata regards the current or the subsequent video stream. All Gamut Metadata Packets within a single video field (VSYNC active edge to VSYNC active edge) shall have the same Current_Gamut_Seq_Num field.

If a packet contains metadata for the currently transmitted video, Affected_Gamut_Seq_Num will be equal to Current_Gamut_Seq_Num. If the packet regards upcoming video, the Affected_Gamut_Seq_Num will be Current_Gamut_Seq_Num + 1 (mod 16). The field Affected_Gamut_Seq_Num shall never be beyond Current_Gamut_Seq_Num + 1, therefore, only current and next gamut may be described.

If it is known by the Source that a packet contains metadata that will be effective for the next video field then the Next_Field flag shall be set, whether or not the metadata is effective for the current video field.

Gamut metadata associated with upcoming video may be transmitted even when the current video has no associated metadata. In this situation, all Gamut Metadata packets transmitted shall indicate that the current stream has no associated metadata (i.e. colorimetry and gamut are described by the currently-valid AVI InfoFrame) by setting No_Current_GBD to 1. At least one such Gamut Metadata packet shall be transmitted in the VBLANK period. All Gamut Metadata packets within that same video field can carry metadata for the upcoming video.

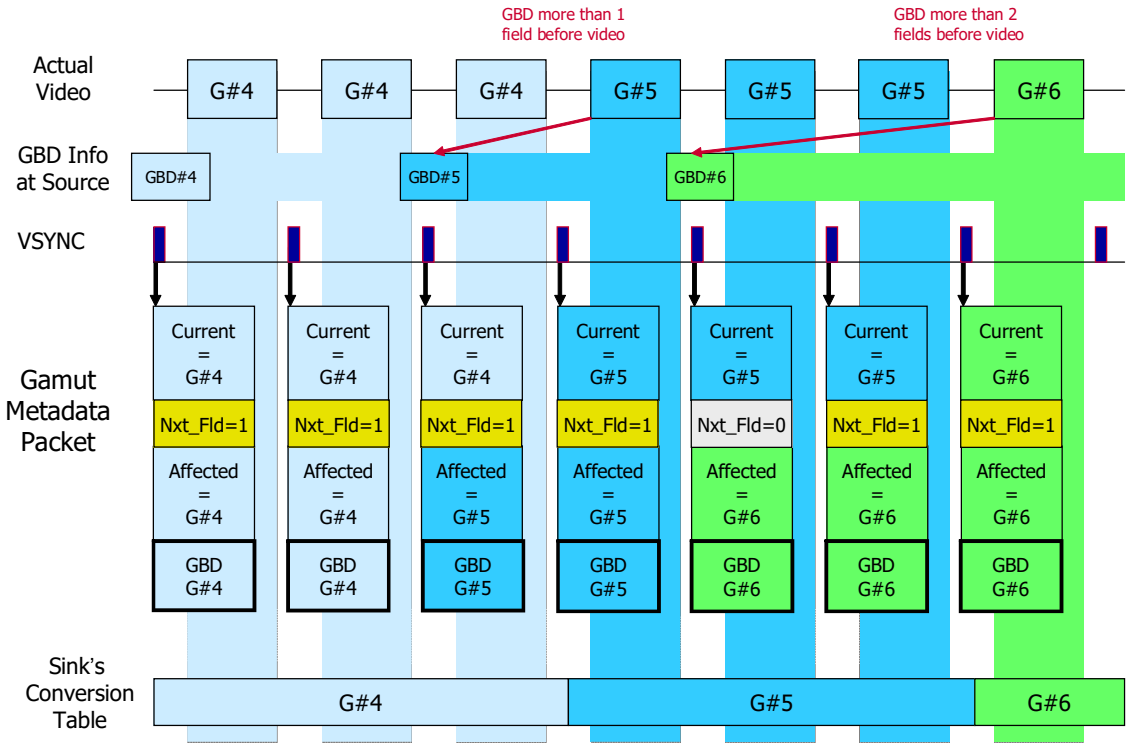


Figure 5-6 Example P0 Transmission Sequence

During the transmission of any video stream that is accompanied by or requires gamut metadata, at least one Gamut Metadata Packet containing a P0 transmission profile GBD shall be transmitted during each VBLANK. This transmission shall occur before the end of the first (VBLANK) video line following the active edge of VSYNC. If no Gamut Metadata Packet is transmitted during this period, then the colorimetry and gamut of the subsequent VACTIVE for that video field shall correspond to that described by the transmitted AVI InfoFrame.

If the Sink indicates support for P2 or higher transmission profiles then the Source may simultaneously transmit two GBDs. In this case, within each video field, the Source may transmit: a P0 profile containing the GBD for the current video and a portion of a higher profile containing the GBD for either the current or the upcoming video. Alternatively, the two simultaneous GBDs may be a P0 profile describing the current video and a P0 profile describing the upcoming video.

Table 5-30 Gamut Metadata Packet Header

Byte \ Bit #	7	6	5	4	3	2	1	0
HB0	0	0	0	0	1	0	1	0
HB1	Next_Field	GBD_profile			Affected_Gamut_Seq_Num			
HB2	No_Crnt _GBD	Rsvd (0)	Packet_Seq		Current_Gamut_Seq_Num			

- Next_Field [1 bit] Set to indicate that the GBD carried in this packet will be effective on the next video field. Specifically, the Affected_Gamut_Seq_Num for this packet will be equal to the Current_Gamut_Seq_Num for the next field. Next_Field should be set even if the GBD is already effective (e.g. Current=Affected).
- No_Current_GBD [1 bit] Set to indicate that there is no gamut metadata available for the currently transmitted video (i.e. current video has a standard colorimetry not requiring a GBD). When set, the field Current_Gamut_Seq_Num is meaningless and shall be ignored by the Sink.
- GBD_profile [3 bits] Transmission profile number:
  0: P0
  1: P1
  2: P2
  3: P3
  other values: reserved.
- Affected_Gamut_Seq_Num [4 bits] Indicates which video fields are relevant for this metadata.
- Current_Gamut_Seq_Num [4 bits] Indicates the gamut number of the currently transmitted video stream. All Gamut Metadata Packets transmitted within the same video field shall have the same Current_Gamut_Seq_Num, even if the Affected_Gamut_Seq_Num varies among the packets.
- Packet_Seq [2 bits] Indicates whether this packet is the only, the first, an intermediate or the last packet in a Gamut Metadata packet sequence.
  = 0 (0b00) Intermediate packet in sequence
  = 1 (0b01) First packet in sequence
  = 2 (0b10) Last packet in sequence
  = 3 (0b11) Only packet in sequence (i.e. P0)

The Gamut Metadata Packet Body differs depending upon transmission profile and whether the packet is the first of a sequence or one of the remaining packets in that sequence.

The packet body for a P0 transmission is defined in Table 5-31.

Table 5-31 Gamut Metadata Packet for P0 Transmission Profile

Packet Byte #	7	6	5	4	3	2	1	0
PB00-PB27	GBD bytes 0 through 27							

When transmitting any GBD requiring more than one Gamut Metadata Packet for transmission, a packet containing the packet body shown in Table 5-32 shall be used. Subsequent packets in that sequence shall use the packet body shown in Table 5-33. If the GBD ends midway through a packet, the rest of the body shall be filled with zeroes by the Source and shall be ignored by the Sink.

Table 5-32 Gamut Metadata Packet for P1 and Higher – 1st Packet of Sequence

Packet Byte #	7	6	5	4	3	2	1	0
PB0	GBD_Length_H							
PB1	GBD_Length_L							
PB2	Checksum							
PB3-PB27	GBD bytes 0 through 24							

- GBD_Length(_H, _L) [2 bytes] Total length (in bytes) of gamut metadata, not including GBD_Length or Checksum.
- Checksum [1 byte] Checksum of every byte covered by GBD_Length field.

Table 5-33 Gamut Metadata Packet for P1 and Higher – Remaining Packets

Packet Byte #	7	6	5	4	3	2	1	0
PB00-PB27	Next 28 bytes of GBD							

5.4 Encoding

5.4.1 Serialization

The stream of TMDS characters produced by the encoder is serialized for transmission on the TMDS data channel. In the discussions that follow, the least significant bit of each character (q_out[0]) is the first bit to be transmitted and the most significant bit (q_out[9]) is the last.

5.4.2 Control Period Coding

Each TMDS channel has two control signals, which are encoded into 10 bits during Control Periods. For each of the three channels these signals are shown in Table 5-34.

Table 5-34 Control-signal Assignment

TMDS Channel	D0	D1
0	HSYNC	VSYNC
1	CTL0	CTL1
2	CTL2	CTL3

The two Control signals for each of the three TMDS channels are encoded as follows:

```
case (D1, D0):
       0, 0:  q_out[9:0] = 0b1101010100;
       0, 1:  q_out[9:0] = 0b0010101011;
       1, 0:  q_out[9:0] = 0b0101010100;
       1, 1:  q_out[9:0] = 0b1010101011;
endcase;
```

5.4.3 TERC4 Coding

TMDS Error Reduction Coding (TERC4) is used during the Data Island period to encode 4 bits per channel into the 10 bits serialized and transmitted.

```
case (D3, D2, D1, D0):
    0000:  q_out[9:0] = 0b1010011100;
    0001:  q_out[9:0] = 0b1001100011;
    0010:  q_out[9:0] = 0b1011100100;
    0011:  q_out[9:0] = 0b1011100010;
    0100:  q_out[9:0] = 0b0101110001;
    0101:  q_out[9:0] = 0b0100011110;
    0110:  q_out[9:0] = 0b0110001110;
    0111:  q_out[9:0] = 0b0100111100;
    1000:  q_out[9:0] = 0b1011001100;
    1001:  q_out[9:0] = 0b0100111001;
    1010:  q_out[9:0] = 0b0110011100;
    1011:  q_out[9:0] = 0b1011000110;
    1100:  q_out[9:0] = 0b1010001110;
    1101:  q_out[9:0] = 0b1001110001;
    1110:  q_out[9:0] = 0b0101100011;
    1111:  q_out[9:0] = 0b1011000011;
endcase;
```

5.4.4 Video Data Coding

5.4.4.1 Video Data Encoding

The following is a description of the encoding algorithm used during transmission of video data. A detailed description of an encoder is given. Other implementations are possible and are permitted but, given the same sequence of input characters, they are required to produce the same sequence of output (10-bit) characters that is generated by the described encoder.

During video data, where each 10-bit character represents 8 bits of pixel data, the encoded characters provide an approximate DC balance as well as a reduction in the number of transitions in the data stream. The encode process for the active data period can be viewed in two stages. The first stage produces a transition-minimized 9-bit code word from the input 8 bits. The second stage produces a 10-bit code word, the finished TMDS character, which will manage the overall DC balance of the transmitted stream of characters.

The 9-bit code word produced by the first stage of the encoder is made up of an 8-bit representation of the transitions found in the input 8 bits, plus a one-bit flag to indicate which of two methods was used to describe the transitions. In both cases, the LSb of the output matches the LSb of the input. With a starting value established, the remaining 7 bits of the output word is derived from sequential exclusive OR (XOR) or exclusive NOR (XNOR) functions of each bit of the input with the previously derived bit. The choice between XOR and XNOR logic is made such that the encoded values contain the fewest possible transitions, and the ninth bit of the code word is used to indicate whether XOR or XNOR functions were used to derive the output code word. The decode of this 9-bit code word is simply a matter of applying either XOR or XNOR gates to the adjacent bits of the code, with the LSb passing from decoder input to decoder output unchanged.

The second stage of the encoder performs an approximate DC balance on the transmitted stream by selectively inverting the 8 data bits of the 9-bit code words produced by the first stage. A tenth bit is added to the code word, to indicate when the inversion has been made. The encoder determines when to invert the next character based on the running disparity between ones and zeros that it tracks in the transmitted stream, and the number of ones and zeros found in the current code word. If too many ones have been transmitted and the input contains more ones than zeros, the code word is inverted. This dynamic encoding decision at the Source is simply decoded at the Sink by the conditional inversion of the input code word based on the tenth bit of the TMDS character. The TMDS code mapping is specified by Figure 5-7 with the definitions of Table 5-35. The encoder produces one of 460 unique 10-bit characters. The encoder shall not generate any other 10-bit character during a Video Data Period.

Upon entering a Video Data Period, the data stream disparity (cnt) shall be considered to be zero by the encoder.

Table 5-35 Encoding Algorithm Definitions

D	The encoder input data set. D is 8-bit pixel data
cnt	This is a register used to keep track of the data stream disparity. A positive value represents the excess number of “1”s that have been transmitted. A negative value represents the excess number of “0”s that have been transmitted. The expression cnt{t-1} indicates the previous value of the disparity for the previous set of input data. The expression cnt(t) indicates the new disparity setting for the current set of input data.
q_m	Intermediate value.
q_out	These 10 bits are the encoded output value.
N_1{x}	This operator returns the number of “1”s in argument “x”
N_0{x}	This operator returns the number of “0”s in argument “x”

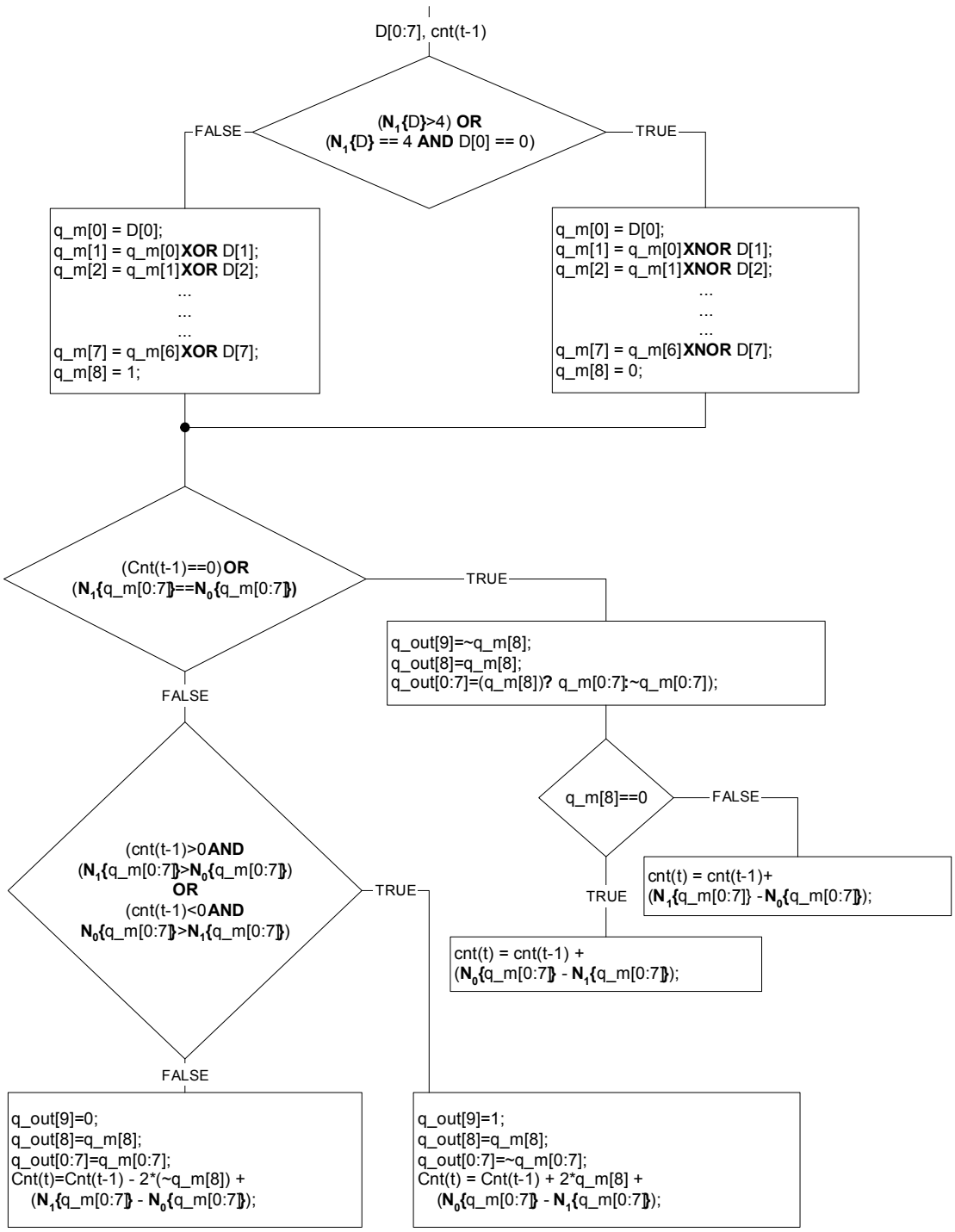


Figure 5-7 TMDS Video Data Encode Algorithm

5.4.4.2 Video Data Decoding

The TMDS decode mapping is specified by Figure 5-8. Alternative implementations are possible but, given the same input data stream, they are required to generate the same output data stream as the described decoder algorithm.

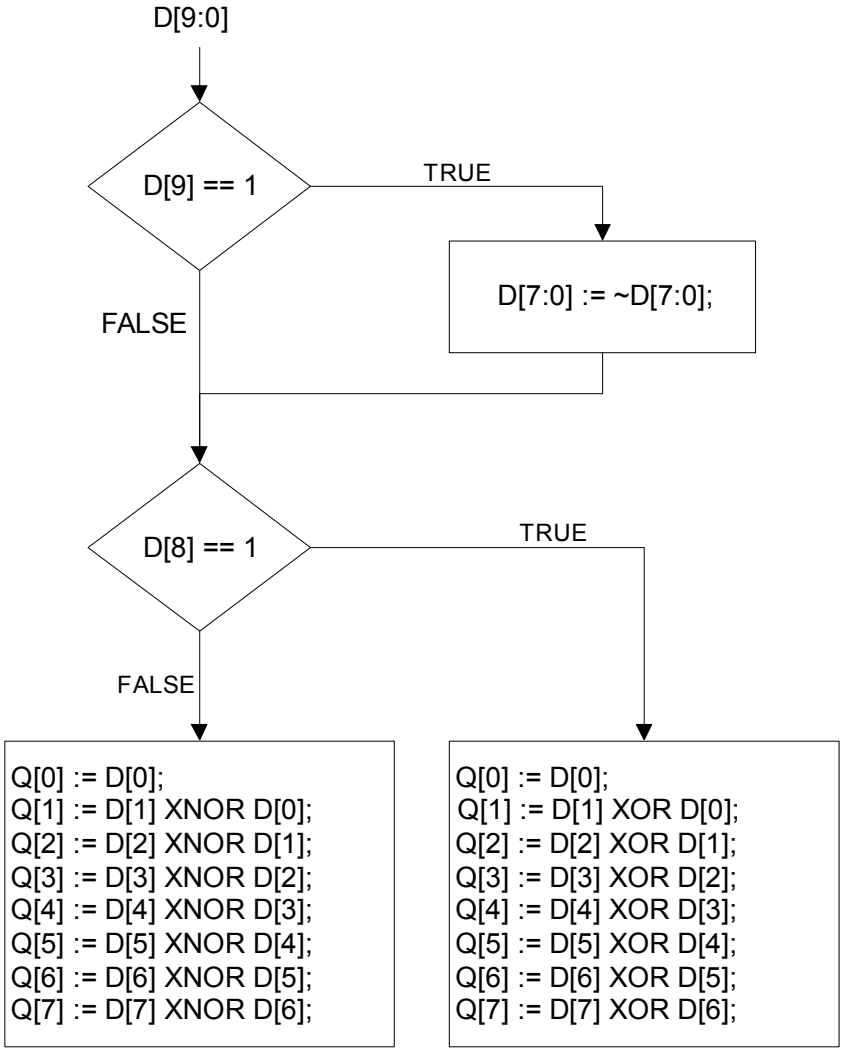


Figure 5-8 TMDS Video Decode Algorithm

6 Video

6.1 Overview

HDMI allows any video format timing to be transmitted and displayed. To maximize interoperability between products, common DTV formats have been defined. These video format timings define the pixel and line counts and timing, synchronization pulse position and duration, and whether the format is interlaced or progressive. HDMI also allows vendor-specific formats to be used.

The video pixels carried across the link shall be in one of three different pixel encodings: RGB 4:4:4, YC_BC_R 4:4:4 or YC_BC_R 4:2:2.

The HDMI Source determines the pixel encoding and video format of the transmitted signal based on the characteristics of the source video, the format and pixel encoding conversions possible at the Source, and the format and pixel encoding capabilities and preferences of the Sink.

6.2 Video Format Support

In order to provide maximum compatibility between video Sources and Sinks, specific minimum requirements have been specified for Sources and Sinks.

6.2.1 Format Support Requirements

Some of the following support requirements are in addition to those specified in CEA-861-D.

- An HDMI Source shall support at least one of the following video format timings:
  - 640x480p @ 59.94/60Hz
  - 720x480p @ 59.94/60Hz
  - 720x576p @ 50Hz
- An HDMI Source that is capable of transmitting any of the following video format timings using any other component analog or uncompressed digital video output, shall be capable of transmitting that video format timing across the HDMI interface.
  - 1280x720p @ 59.94/60Hz
  - 1920x1080i @ 59.94/60Hz
  - 720x480p @ 59.94/60Hz
  - 1280x720p @ 50Hz
  - 1920x1080i @ 50Hz
  - 720x576p @ 50Hz

- An HDMI Sink that accepts 60Hz video formats shall support the 640x480p @ 59.94/60Hz and 720x480p @ 59.94/60Hz video format timings.
- An HDMI Sink that accepts 50Hz video formats shall support the 640x480p @ 59.94/60Hz and 720x576p @ 50Hz video format timings.

- An HDMI Sink that accepts 60Hz video formats, and that supports HDTV capability, shall support 1280x720p @ 59.94/60Hz or 1920x1080i @ 59.94/60Hz video format timings.
- An HDMI Sink that accepts 50Hz video formats, and that supports HDTV capability, shall support 1280x720p @ 50Hz or 1920x1080i @ 50Hz video format timings.
- An HDMI Sink that is capable of receiving any of the following video format timings using any other component analog or uncompressed digital video input, shall be capable of receiving that format across the HDMI interface.
  - 1280x720p @ 59.94/60Hz
  - 1920x1080i @ 59.94/60Hz
  - 1280x720p @ 50Hz
  - 1920x1080i @ 50Hz

Additional recommendations for video format handling by Sources and Sinks is given in Appendix F.

6.2.2 Video Control Signals : HSYNC, VSYNC

During the Data Island period, HDMI carries HSYNC and VSYNC signals using encoded bits on Channel 0. During Video Data periods, HDMI does not carry HSYNC and VSYNC and the Sink should assume that these signals remain constant. During Control periods, HDMI carries HSYNC and VSYNC signals through the use of four different control characters on TMDS Channel 0.

6.2.3 Pixel Encoding Requirements

Only pixel encodings of RGB 4:4:4, YC_BC_R 4:2:2, and YC_BC_R 4:4:4 (as specified in Section 6.5) may be used on HDMI.

All HDMI Sources and Sinks shall be capable of supporting RGB 4:4:4 pixel encoding.

All HDMI Sources shall support either YC_BC_R 4:2:2 or YC_BC_R 4:4:4 pixel encoding whenever that device is capable of transmitting a color-difference color space across any other component analog or digital video interface except where that device would be required to convert RGB video to YC_BC_R in order to meet this requirement.

All HDMI Sinks shall be capable of supporting both YC_BC_R 4:4:4 and YC_BC_R 4:2:2 pixel encoding when that device is capable of supporting a color-difference color space from any other component analog or digital video input.

If an HDMI Sink supports either YC_BC_R 4:2:2 or YC_BC_R 4:4:4 then both shall be supported.

An HDMI Source may determine the pixel-encodings that are supported by the Sink through the use of the E-EDID. If the Sink indicates that it supports YC_BC_R-formatted video data and if the Source can deliver YC_BC_R data, then it can enable the transfer of this data across the link.

6.2.4 Color Depth Requirements

HDMI Sources and Sinks may support color depths of 24, 30, 36 and/or 48 bits per pixel. All HDMI Sources and Sinks shall support 24 bits per pixel.

Color depths greater than 24 bits are defined to be “Deep Color” modes. All Deep Color modes are optional though if an HDMI Source or Sink supports any Deep Color mode, it shall support 36-bit mode.

For each supported Deep Color mode, RGB 4:4:4 shall be supported and optionally YC_BC_R 4:4:4 may be supported.

YC_BC_R 4:2:2 is not permitted for any Deep Color mode.

An HDMI Sink shall support all EDID-indicated Deep Color modes on all EDID-indicated video formats except if that combination exceeds the Max_TMDS_Clock indication.

An HDMI Source shall not send any Deep Color mode to a Sink that does not indicate support for that mode.

6.3 Video Format Timing Specifications

All specified video line pixel counts and video field line counts (both active and total) and HSYNC and VSYNC positions, polarities, and durations shall be adhered to when transmitting a specified video format timing.

For example, if a Source is processing material with fewer active pixels per line than required (i.e. 704 pixels vs. 720 pixels for standard definition MPEG2 material), it may add pixels to the left and right of the supplied material before transmitting across HDMI. AVI bar info may need to be adjusted to account for these added pixels.

Detailed timing is found in CEA-861-D or a later version of CEA-861 for the following video format timings.

6.3.1 Primary Video Format Timings

- 640x480p @ 59.94/60Hz
- 1280x720p @ 59.94/60Hz
- 1920x1080i @ 59.94/60Hz
- 720x480p @ 59.94/60Hz
- 720(1440)x480i @ 59.94/60Hz
- 1280x720p @ 50Hz
- 1920x1080i @ 50Hz
- 720x576p @ 50Hz
- 720(1440)x576i @ 50Hz

6.3.2 Secondary Video Format Timings

- 720(1440)x240p @ 59.94/60Hz
- 2880x480i @ 59.94/60Hz
- 2880x240p @ 59.94/60Hz

- 1440x480p @ 59.94/60Hz
- 1920x1080p @ 59.94/60Hz
- 720(1440)x288p @ 50Hz
- 2880x576i @ 50Hz
- 2880x288p @ 50Hz
- 1440x576p @ 50Hz
- 1920x1080p @ 50Hz
- 1920x1080p @ 23.98/24Hz
- 1920x1080p @ 25Hz
- 1920x1080p @ 29.97/30Hz
- 2880x480p @ 59.94/60Hz
- 2880x576p @ 50Hz
- 1920x1080i (1250 total) @ 50Hz
- 720(1440)x480i @ 119.88/120Hz
- 720x480p @ 119.88/120Hz
- 1920x1080i @ 119.88/120Hz
- 1280x720p @ 119.88/120Hz
- 720(1440)x480i @ 239.76/240Hz
- 720x480p @ 239.76/240Hz
- 720(1440)x576i @ 100Hz
- 720x576p @ 100Hz
- 1920x1080i @ 100Hz
- 1280x720p @ 100Hz
- 720(1440)x576i @ 200Hz
- 720X576p @ 200Hz

6.4 Pixel-Repetition

Video formats with native pixel rates below 25 Mpixels/sec require pixel-repetition in order to be carried across a TMDS link. 720x480i and 720x576i video format timings shall always be pixel-repeated.

The HDMI Source indicates the use of pixel-repetition with the Pixel Repetition (PR0:PR3) field in the AVI InfoFrame. This field indicates to the HDMI Sink how many repetitions of each unique pixel are transmitted. In non-repeated formats, this value is zero.

For pixel-repeated formats, this value indicates the number of pixels that may be discarded by the Sink without losing real image content.

The Source shall always accurately indicate the pixel repetition count being used. The use of the Pixel Repetition field is optional for HDMI Sink.

The use of this pixel-repetition count field is more fully described in CEA-861-D.

6.5 Pixel Encodings and Color Depth

There are three different pixel encodings that may be sent across an HDMI cable: YC_BC_R 4:4:4, YC_BC_R 4:2:2 and RGB 4:4:4. Whichever encoding is used, it shall conform to one of the methods described in this section.

There are four color depths supported: 24-, 30-, 36- and 48-bits per pixel. Only RGB 4:4:4 and YC_BC_R 4:4:4 are permitted at depths greater than 24-bits ("Deep Color" modes).

6.5.1 24-Bit Pixel Encodings

Figure 6-1 shows the default encoding, RGB 4:4:4 for 24-bit color depth. The R, G, and B components of the first pixel for a given line of video are transferred on the first pixel of the video data period following the Guard Band characters.

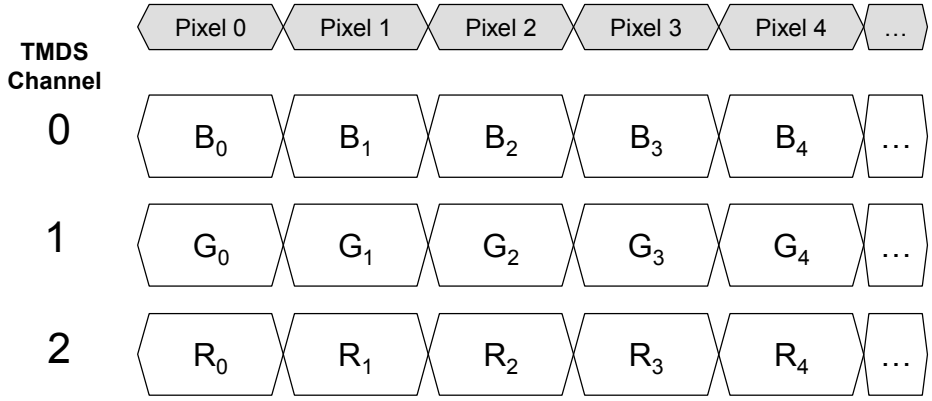


Figure 6-1 Default pixel encoding: RGB 4:4:4, 8 bits/component

Figure 6-2 shows the signal mapping and timing for transferring 24-bit YC_BC_R 4:2:2 data across HDMI. Because 4:2:2 data only requires two components per pixel, more bits are allocated per component. The available 24 bits are split into 12 bits for the Y component and 12 bits for the C components.

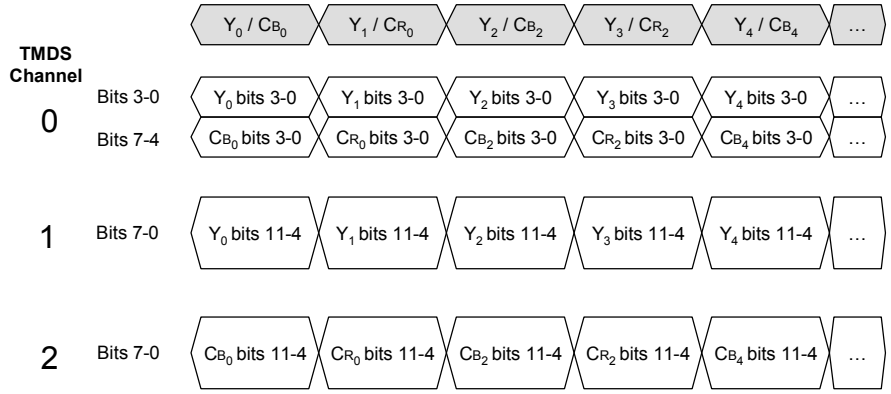


Figure 6-2 YC_BC_R 4:2:2 component

The YC_BC_R 4:2:2 pixel encoding on HDMI closely resembles standard ITU-R BT.601. The high-order 8 bits of the Y samples are mapped onto the 8 bits of Channel 1 and the low-order 4 bits are mapped onto the low-order 4 bits of Channel 0. If fewer than 12 bits are used, the valid bits shall be left-justified (i.e. MSb=MSb) with zeroes padding the bits below the LSb.

The first pixel transmitted within a Video Data Period contains three components, Y0, Cb0 and Cr0. The Y0 and Cb0 components are transmitted during the first pixel period while Cr0 is transmitted during the second pixel period. This second pixel period also contains the only component for the second pixel – Y1. In this way, the link carries one C_B sample for every two pixels and one Cr sample for every two pixels. These two components (C_B and C_R) are multiplexed onto the same signal paths on the link.

At the third pixel, this process is repeated with the Y and C_B components for the third pixel being transmitted, followed, in the next pixel period, by the C_R component of the third pixel and the Y component of the fourth pixel.

YC_BC_R 4:4:4 data is transferred using the scheme illustrated in Figure 6-3.

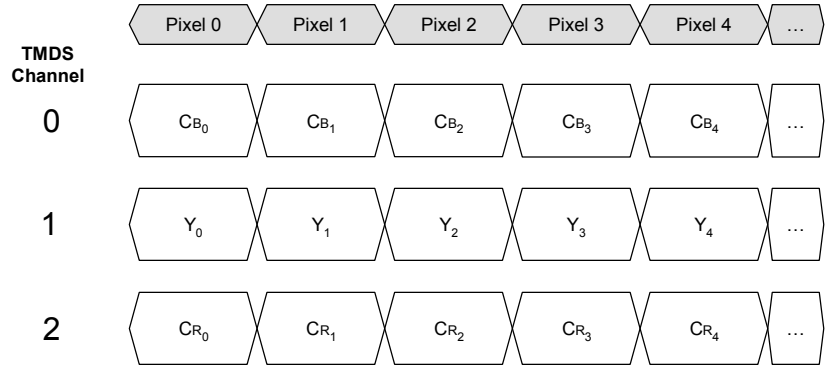


Figure 6-3 8-bit YC_BC_R 4:4:4 mapping

6.5.2 Deep Color Pixel Packing

For a color depth of 24 bits/pixel, pixels are carried at a rate of one pixel per TMDS clock. At deeper color depths, the TMDS clock is run faster than the source pixel clock providing the extra bandwidth for the additional bits. The TMDS clock rate is increased by the ratio of the pixel size to 24 bits:

- 24 bit mode: TMDS clock = 1.0 x pixel clock (1:1)
- 30 bit mode: TMDS clock = 1.25 x pixel clock (5:4)
- 36 bit mode: TMDS clock = 1.5 x pixel clock (3:2)
- 48 bit mode: TMDS clock = 2.0 x pixel clock (2:1)

When operating in a Deep Color mode, all video data (pixels) and signaling (HSYNC, VSYNC, DE transitions) are grouped into a series of packed Pixel Groups, each carrying the same number of pixels and each requiring the same number of TMDS clocks for transmission. On each TMDS

clock, one Fragment of the Pixel Group is transmitted. The number of pixels per group and number of fragments per group depends on the pixel size:

- 24 bit mode: 1 pixel/group, 1 fragment/group
- 30 bit mode: 4 pixels/group, 5 fragments/group
- 36 bit mode: 2 pixels/group, 3 fragments/group
- 48 bit mode: 1 pixel/group, 2 fragments/group

During active video, the input pixel data is packed into these groups. During blanking, HSYNC and VSYNC are packed into these same groups. In this way, all video-related protocol elements are carried at a direct ratio to the pixel clock, thus ensuring no change to the relationship between the pixel clock and the pixel data, DE transitions and HSYNC or VSYNC transitions. This also allows any sequence of HSYNC, VSYNC, DE transitions, etc. that can be supported at 24 bits/pixel to be supported equally in any other pixel size.

All other HDMI protocol elements are unaffected by the Deep Color pixel packing. Data Islands, Video Guard Bands and Preambles occur as they do in normal (24-bit) mode – each Preamble is 8 TMDS clocks, each Data Island packet is 32 TMDS clocks, and each Guard Band is 2 TMDS clocks.

As shown above, a pixel group consists of 1, 2, or 4 pixels. Each pixel group is broken into 1, 2, 3 or 5 pixel fragments transmitted one fragment per TMDS clock.

Each TMDS character period (one TMDS clock) in the transmitted stream carries a single Fragment of a Pixel Group and so represents a particular Packing Phase of the group. It is necessary for the Sink to determine which character in the stream of characters represents the start of a new group, or phase 0, in order to synchronize its pixel unpacking state with the source's pixel packing state. To accomplish this, the source sends a packet indicating the packing phase of a specific pixel (see 6.5.3 for packet details). This packet is sent at least once per video field indicating the then-current packing phase. The sink uses this data to initially determine where each new group starts should also use this periodic update to verify that it is still synchronized or to recover from gross errors on the link.

The following tables specify all Pixel Encodings for all color depths. For each mode, the packing for each phase is described. Packing phases for active video are identified as "*m*P*n*" (10P0, 10P1, etc. while the packing phases for blanking are identified as "*m*C*n*" (10C0, 10C1, etc.).

24 bit mode: P (pixels/group) = 1 pixel; L (fragments/group) = 1 fragment (1 TMDS character). Standard HDMI format.

Fragment	Phase	Pixels	8 bit HDMI pixel data code (to encoder)							
			Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
8P0	0	A	A0	A1	A2	A3	A4	A5	A6	A7

30 bit mode: P = 4 pixels; L = 5 fragments

Fragment	Phase	Pixels	8 bit HDMI pixel data code (to encoder)							
			Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
10P0	0	A	A0	A1	A2	A3	A4	A5	A6	A7
10P1	1	A+B	A8	A9	B0	B1	B2	B3	B4	B5
10P2	2	B+C	B6	B7	B8	B9	C0	C1	C2	C3
10P3	3	C+D	C4	C5	C6	C7	C8	C9	D0	D1
10P4	4	D	D2	D3	D4	D5	D6	D7	D8	D9

36 bit mode: P = 2 pixels; L = 3 fragments

Fragment	Phase	Pixels	8 bit HDMI pixel data code (to encoder)							
			Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
12P0	0	A	A0	A1	A2	A3	A4	A5	A6	A7
12P1	1	A+B	A8	A9	A10	A11	B0	B1	B2	B3
12P2	2	B	B4	B5	B6	B7	B8	B9	B10	B11

48 bit mode: P = 1 pixel; L = 2 fragments

Fragment	Phase	Pixels	8 bit HDMI pixel data code (to encoder)							
			Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
16P0	0	A	A0	A1	A2	A3	A4	A5	A6	A7
16P1	1	A	A8	A9	A10	A11	A12	A13	A14	A15

During blanking, one HSYNC and VSYNC value per pixel clock is carried in each Pixel Group. This provides one more HSYNC and VSYNC slot per group than is required (e.g. 5 TMDS clocks for 4 pixels) so the HSYNC and VSYNC values are simply repeated on the last TMDS clock of a group.

The following tables specify, for each mode, the group size and the sequence of HSYNC and VSYNC transmission within a group.

Source HSYNC/VSYNC values for each pixel are labeled S, T, U, V (as needed). Source HSYNC/VSYNC value S is the leftmost (earliest) code in the group.

- 24 bit mode:

Group size = 1 pixel; 1 fragment.

Fragment	HS/VS value
8C0	S

- 30 bit mode:

In 30-bit mode, if the Video Data Period ends before the pixel group boundary, the remaining fragments are filled using one or more steps from the 10PC*n* sequence listed in the table below until the group boundary is reached (step 10PC4). After that, the normal sequence is used (steps 10C*n*).

Group size = 4 pixels; 5 fragments.

Fragment	HS/VS value
10C0	S
10C1	T
10C2	U

10C3	V
10C4	V

30-bit mode remnant (Falling Edge of DE occurs mid-group). Bridge states for transition from 10P*n* to 10C0 are named “10PC*n*”:

Fragment	HS/VS value
10PC2	T
10PC3	U
10PC4	V

- 36 bit mode:

Group size = 2 pixels; 3 fragment.

Fragment	HS/VS value
12C0	S
12C1	T
12C2	T

- 48 bit mode:

Group size = 1 pixel; 2 fragment.

Fragment	HS/VS value
16C0	S
16C1	S

6.5.3 Deep Color Mode / Phase Indication

When in a Deep Color mode, the Source and Sink each records the packing phase of the last pixel character of a Video Data period. The Source occasionally sends a General Control Packet (GCP) communicating the current color depth and the packing phase of the last pixel character sent prior to the GCP. This data is valid in the GCP whenever CD (CD0, CD1, CD2, CD3) is non-zero. Whenever the Sink receives a GCP with non-zero CD data, it should compare the receiver’s own color depth and phase with the CD data. If they do not match, the Sink should adjust its color depth and/or phase to match the CD data.

When transmitting Deep Color, the Source shall send a General Control Packet (GCP) with an accurate CD field indicating the current color depth and with the PP field (PP0, PP1, PP2, PP3) indicating the packing phase of the last pixel character (within the last Video Data Period) sent prior to the GCP. Sources shall only send GCPs with non-zero CD to Sinks that indicate support for Deep Color, and shall only select color depths supported by the Sink.

Once a Source sends a GCP with non-zero CD to a sink, it should continue sending GCPs with non-zero CD at least once per video field even if reverting to 24-bit color, as long as the Sink continues to support Deep Color.

If the Sink does not receive a GCP with non-zero CD for more than 4 consecutive video fields, it should exit deep color mode (revert to 24-bit color).

Color Depth field (CC) of SB1:

- When CD is zero, no information about color depth is indicated. PP shall be zero.
- When CD is non-zero, the color depth is indicated and the packing phase bits (PP) are valid.

Table 6-1 Color Depth (CD field) Values

CD3	CD2	CD1	CD0	Color Depth
0	0	0	0	Color Depth not indicated
0	0	0	1	Reserved
0	0	1	0	Reserved
0	0	1	1	Reserved
0	1	0	0	24 bits per pixel
0	1	0	1	30 bits per pixel
0	1	1	0	36 bits per pixel
0	1	1	1	48 bits per pixel
All other values				Reserved

A CD field of zero (Color Depth not indicated) shall be used whenever the Sink does not indicate support for Deep Color. [pw130]This value may also be used in Deep Color mode to transmit a GCP indicating only non-Deep Color information (e.g. AVMUTE).

When the CD field indicates 24 bits per pixel, the PP field is invalid and should be ignored by the Sink.

Pixel Packing Phase field (PP) of SB1:

- When the CD field is zero, the PP field shall also be zero.
- When the CD field is non-zero, the PP field indicates the packing phase of the last fragment in the most recent Video Data Period (prior to the GCP message).

Table 6-2 shows the specific PP values for each of the packing phases shown in the packing phase tables earlier.

Table 6-2 Pixel Packing Phase (PP field) Values

PP3	PP2	PP1	PP0	Pixel packing Phase
0	0	0	0	Phase 4 (10P4)
0	0	0	1	Phase 1 (10P1, 12P1, 16P1)
0	0	1	0	Phase 2 (10P2, 12P2)
0	0	1	1	Phase 3 (10P3)
0	1	0	0	Reserved
All other values				Reserved

Packing phase 0 does not need to be indicated using the PP bits. This is not necessary as phase 0 always represents only part of the first pixel of the group and therefore, no Video Data Period will end at phase 0. If the active video ended after the first pixel, then the final phase will be phase 1, containing the last bits of the first pixel.

If the transmitted video format has timing such that the phase of the first pixel of every Video Data Period corresponds to pixel packing phase 0 (e.g. 10P0, 12P0, 16P0), the Source may set the Default_Phase bit in the GCP. The Sink may use this bit to optimize it's filtering or handling of the PP field.

Default_Phase field of GCP SB2:

- When Default_Phase is 0, the PP bits may vary and the first pixel of each Video Data Period may vary.
- When Default_Phase is 1, the following will be true:
  - The first pixel of each Video Data Period shall always have a pixel packing phase of 0 (10P0, 12P0, 16P0).
  - The first pixel following each Video Data Period shall have a pixel packing phase of 0 (10C0, 12C0, 16C0).
  - The PP bits shall be constant for all GCPs and will be equal to the last packing phase (10P4, 12P2, 16P1).
  - The first pixel following every transition of HSYNC or VSYNC shall have a pixel packing phase of 0 (10C0, 12C0, 16C0).

6.5.4 Pixel Repetition

During pixel-doubling (Pixel_Repetition_Count = 1), all of the data sent across during the first pixel period will be repeated during the second pixel period. The third pixel period will then represent the second actual pixel and so on. This is shown below for YC_BC_R 4:2:2.

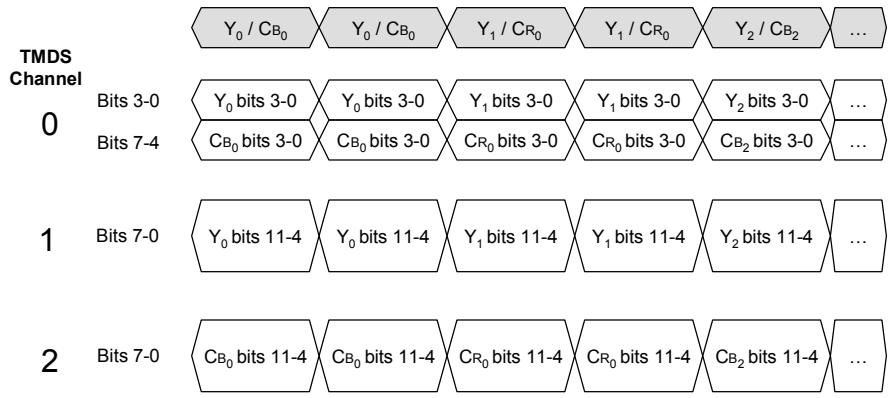


Figure 6-4 YC_BC_R 4:2:2 with Pixel-Doubling

Pixel repetition is permitted in conjunction with Deep Color modes. The Source replicates the pixels as described above prior to Deep Color packing into multiple fragments.

6.6 Video Quantization Ranges

Black and white levels for video components shall be either “Full Range” or “Limited Range.” YC_BC_R components shall always be Limited Range while RGB components may be either Full Range or Limited Range. While using RGB, Limited Range shall be used for all video formats defined in CEA-861-D, with the exception of VGA (640x480) format, which requires Full Range.

Table 6-3 Video Color Component Ranges

Color Components	Component Bit Depth	for Full range		for Limited range	
		Black level	Nominal Peak (White level)	Black level	Nominal Peak (White level)
R / G / B	8	0	255	16	235
R / G / B	10	0	1023	64	940
R / G / B	12	0	4095	256	3760
R / G / B	16	0	65535	4096	60160

Color Component	Bit Depth	for Full range	for Limited range		
			Black level	Nominal Peak	Valid range
Y	8	not allowed	16	235	1 to 254
C_B / C_R			-	16 and 240	
Y	10	not allowed	64	940	4 to 1019
C_B / C_R			-	64 and 960	
Y	12	not allowed	256	3760	16 to 4079
C_B / C_R			-	256 and 3840	
Y	16	not allowed	4096	60160	256 to 65279
C_B / C_R			-	4096 and 61440	

For component ranges for xvYCC, please refer to IEC 61966-2-4.

6.7 Colorimetry

6.7.1 Default Colorimetry

Sources will typically use the specific default colorimetry for the video format being transmitted. If no colorimetry is indicated in the AVI InfoFrame's C field (C1, C0) then the colorimetry of the transmitted signal shall match the default colorimetry for the transmitted video format.

6.7.1.1 480p, 480i, 576p, 576i, 240p and 288p

The default colorimetry for all 480-line, 576-line, 240-line, and 288-line video formats described in CEA-861-D is based on SMPTE 170M.

6.7.1.2 1080i, 1080p and 720p

The default colorimetry for high-definition video formats (1080i, 1080p and 720p) described in CEA-861-D is based on ITU-R BT.709-5.

6.7.1.3 Other Formats

The default colorimetry of other video formats is sRGB.

6.7.2 Applicable Colorimetry Standards

6.7.2.1 SMPTE 170M / ITU-R BT.601

For any video categorized as SMPTE 170M, ITU-R BT.601-5 Section 3.5 shall be used for any color space conversion needed in the course of processing.

The encoding parameter values shall be as defined in Table 3 of ITU-R BT.601-5 and as summarized in Section 6.6.

6.7.2.2 ITU-R BT.709-5

For any video categorized as ITU-R BT.709, Part 1, Section 4 of that document shall be used for any color space conversion needed in the course of processing.

The digital representation shall be as defined in Part 1, Section 6.10 of ITU-R BT.709-5 and as summarized in Section 6.6.

6.7.2.3 IEC 61966-2-4 (xvYCC)

IEC 61966-2-4 defines the "Extended-gamut YCC color space for video applications". It is based on the YCC color encoding described in ITU-R BT.709-5 but extends its definition to a much wider gamut.

$xvYCC_{601}$ is based on the colorimetry defined in ITU-R BT.601, and $xvYCC_{709}$ is based on the colorimetry defined in ITU-R BT.709.

Refer to Chapter 4.3 of IEC 61966-2-4 for more details.

Any Source transmission of xvYCC video (either $xvYCC_{601}$ or $xvYCC_{709}$) shall be accompanied by the transmission of valid gamut boundary metadata.

If the attached Sink does not support xvYCC or does not support Gamut Metadata Packets, then the Source should not transmit xvYCC-encoded video and shall not indicate $xvYCC_{601}$ or $xvYCC_{709}$ in the AVI InfoFrame.

6.7.3 Gamut-Related Metadata

HDMI has the ability to carry a description of the video gamut boundary using the Gamut Metadata Packet.

The Sink indicates support for specific transmission profiles by setting one or more of the MD0, MD1, etc. bits in the Colorimetry Data Block.

If the attached Sink's EDID does not include a Colorimetry Data Block then the Source shall not transmit Gamut Metadata Packets. Note that xvYCC colorimetry requires transmission of the gamut metadata.

7 Audio

7.1 Relationship with IEC 60958/IEC 61937

L-PCM and IEC 61937 compressed audio data is formatted in the Audio Sample Packet or in the High Bitrate Audio Stream Packet as a structure that closely resembles an IEC 60958 frame. (Note: One Bit Audio and DST use a different mechanism – see the overview in sections 7.9.)

On HDMI, each IEC 60958 sub-frame is represented as a 28-bit word. There is no encoding of the preamble type, which instead is replaced with a “B” bit (start-of-block) in each Audio Sample packet. The B bit shall be set for a “B, W” frame and shall be clear for an “M, W” frame. (IEC 60958-1 Section 4.1.2). No other sub-frame preamble combinations are allowed.

Except where specifically indicated in this document, the behavior of all fields within the Audio Sample Subpackets shall follow the corresponding rules specified in the IEC 60958 or IEC 61937 specifications.

HDMI supports any IEC 61937 compressed audio format with a maximum bitrate of 6.144Mbps (frame rate of 192kHz) or less using Layout 0 of the Audio Sample Packet and higher bitrates using the HBR Audio Stream Packet (defined in section 5.3.11). See section 7.6 for more details.

Any IEC 61937-compliant compressed audio format may be supported by an HDMI Source or Sink if a published version of CEA-861 specifies an appropriate Audio Format Code for use in the EDID’s CEA Short Audio Descriptor (see CEA-861-D, table 37).

When receiving multi-channel audio, the Sink should not assume that Channel Status bits carried in Subpackets other than Subpacket 0 will have valid data.

7.2 Audio Sample Clock Capture and Regeneration

Audio data being carried across the HDMI link, which is driven by a TMDS clock running at a rate corresponding to the video pixel rate, does not retain the original audio sample clock. The task of recreating this clock at the Sink is called Audio Clock Regeneration.

There are a variety of clock regeneration methods that can be implemented in an HDMI Sink, each with a different set of performance characteristics. This specification does not attempt to define exactly how these mechanisms operate. It does however present a possible configuration and it does define the data items that the HDMI Source shall supply to the HDMI Sink in order to allow the HDMI Sink to adequately regenerate the audio clock. It also defines how that data shall be generated.

In many video source devices, the audio and video clocks are generated from a common clock (coherent clocks). In this situation, there exists a rational (integer divided by integer) relationship between these two clocks. The HDMI clock regeneration architecture can take advantage of this rational relationship and can also work in an environment where there is no such relationship between these two clocks, that is, where the two clocks are truly asynchronous or where their relationship is unknown.

Figure 7-1 Audio Clock Regeneration model, illustrates the overall system architecture model used by HDMI for audio clock regeneration. The Source shall determine the fractional relationship between the TMDS clock and an audio reference clock (128 * audio sample rate [f_s]) and shall pass the numerator and denominator of that fraction to the Sink across the HDMI link. The Sink

may then recreate the audio clock from the TMDS clock by using a clock divider and a clock multiplier.

The exact relationship between the two clocks will be:

$$128 * f_S = f_{TMDS\_clock} * N / CTS.$$

The Source shall determine the value of the numerator N as specified in Section 7.2.1. Typically, this value N will be used in a clock divider to generate an intermediate clock that is slower than the $128 * f_S$ clock by the factor N. The Source will typically determine the value of the denominator CTS (Cycle Time Stamp) by counting the number of TMDS clocks in each of the $128 * f_S/N$ clocks.

If there is a constant fractional relationship between these two clocks, and the two clocks are exactly synchronous, then the CTS value will quickly come to a constant value. If the clocks are asynchronous, or there is some amount of jitter between the two clocks, then the CTS value will typically alternate between two or three different values. Greater variations are possible with larger jitter.

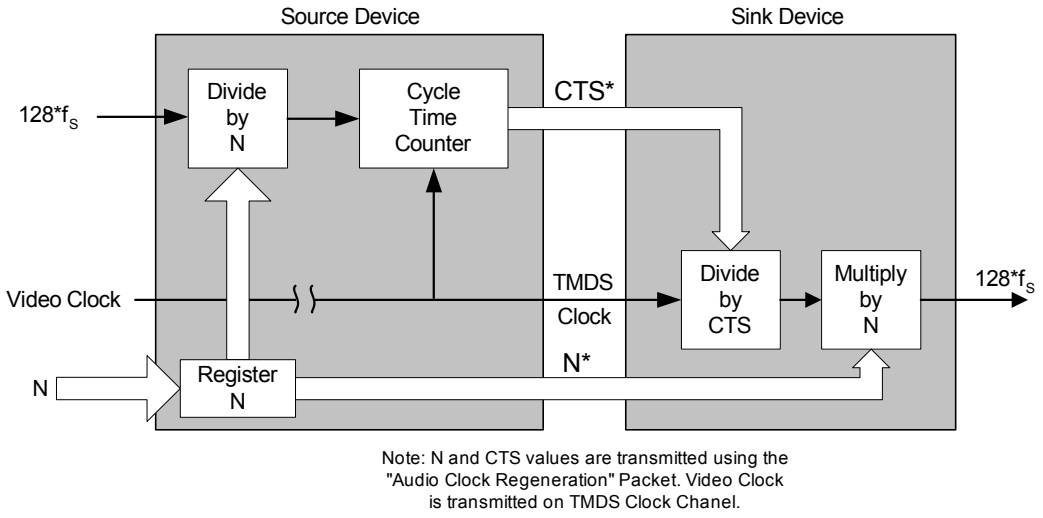


Figure 7-1 Audio Clock Regeneration model

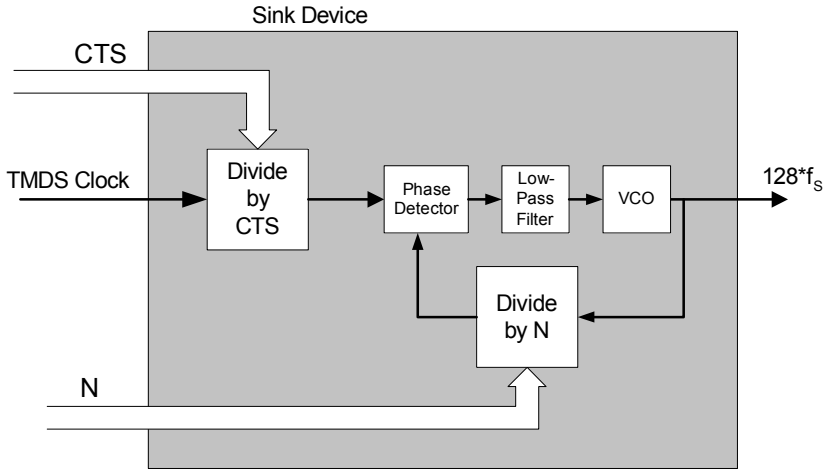


Figure 7-2 Optional Implementation: Audio Sink

It is expected that most Sinks will be implemented with an architecture similar to that shown in Figure 7-2, however, it is permitted and possible to devise an audio clock regeneration function that does not take advantage of the N or CTS values passed to the Sink.

Note that the ACR mechanism uses the TMDS clock, not the pixel clock. When transmitting normal 24-bit pixels, the two are equivalent but when transmitting Deep Color modes, the TMDS clock, used for the ACR, will be running at a higher rate than the pixel clock.

It is recommended that Source devices be designed to support integer CTS values whenever possible.

7.2.1 N parameter

N shall be an integer number and shall meet the following restriction:

$$128 * f_S / 1500\text{Hz} \le N \le 128 * f_S / 300\text{Hz}$$

with a recommended optimal value of

$$128 * f_S / 1000\text{Hz} \text{ approximately equals } N$$

For coherent audio and video clock Sources, the tables in section 7.2.3 below should be used to determine the value of N. For non-coherent Sources or Sources where coherency is not known, the equations above should be used.

7.2.2 CTS parameter

CTS shall be an integer number that satisfies the following:

$$\text{(Average CTS value)} = (f_{TMDS\_clock} * N) / (128 * f_S)$$

7.2.3 Recommended N and Expected CTS Values

The recommended value of N for several standard TMDS clock rates are given in Table 7-1, Table 7-2, and Table 7-3. It is recommended that Sources with non-coherent clocks use the values listed for a TMDS clock of “Other”.

Table 7-1 Recommended N and Expected CTS for 32kHz

	32 kHz	
TMDS Clock (MHz)	N	CTS
25.2 / 1.001	4576	28125
25.2	4096	25200
27	4096	27000
27 * 1.001	4096	27027
54	4096	54000
54 * 1.001	4096	54054
74.25 / 1.001	11648	210937-210938*
74.25	4096	74250
148.5 / 1.001	11648	421875
148.5	4096	148500
Other	4096	Measured

** Note: This value will alternate because of restriction on N.*

Table 7-2 Recommended N and Expected CTS for 44.1kHz and Multiples

	44.1 kHz		88.2 kHz		176.4 kHz	
TMDS Clock (MHz)	N	CTS	N	CTS	N	CTS
25.2 / 1.001	7007	31250	14014	31250	28028	31250
25.2	6272	28000	12544	28000	25088	28000
27	6272	30000	12544	30000	25088	30000
27 * 1.001	6272	30030	12544	30030	25088	30030
54	6272	60000	12544	60000	25088	60000
54 * 1.001	6272	60060	12544	60060	25088	60060
74.25 / 1.001	17836	234375	35672	234375	71344	234375
74.25	6272	82500	12544	82500	25088	82500
148.5 / 1.001	8918	234375	17836	234375	35672	234375
148.5	6272	165000	12544	165000	25088	165000
Other	6272	measured	12544	measured	25088	measured

Table 7-3 Recommended N and Expected CTS for 48kHz and Multiples

TMDS Clock (MHz)	48 kHz		96 kHz		192 kHz	
	N	CTS	N	CTS	N	CTS
25.2 / 1.001	6864	28125	13728	28125	27456	28125
25.2	6144	25200	12288	25200	24576	25200
27	6144	27000	12288	27000	24576	27000
27 * 1.001	6144	27027	12288	27027	24576	27027
54	6144	54000	12288	54000	24576	54000
54 * 1.001	6144	54054	12288	54054	24576	54054
74.25 / 1.001	11648	140625	23296	140625	46592	140625
74.25	6144	74250	12288	74250	24576	74250
148.5 / 1.001	5824	140625	11648	140625	23296	140625
148.5	6144	148500	12288	148500	24576	148500
Other	6144	measured	12288	measured	24576	measured

7.2.4 L-PCM and IEC 61937 Compressed Audio ACR

For any L-PCM stream, the ACR f_S value shall be equal to the audio sample rate.

For any IEC 61937 compressed audio with an IEC 60958 frame rate at or below 192kHz, the ACR f_S value shall be equal to the frame rate. For any such stream with an IEC 60958 frame rate above 192kHz, the ACR f_S value shall be $1/4^{th}$ of the frame rate.

7.2.5 One Bit Audio ACR

For any One Bit Audio stream, the ACR f_S value shall be $1/64^{th}$ of the bit rate. For One Bit Audio data from Super Audio CD (2.8224MHz) the ACR f_S would therefore be 44.1kHz.

7.2.6 DST Audio ACR

For DST audio streams, the ACR f_S corresponds to the sample rate of the underlying compressed audio samples, which is typically 64*44.1kHz (2.8224MHz). This is true whether DST_Normal or DST_Double rate is used.

7.3 Audio Sample Rates and Support Requirements

If an HDMI Source supports audio transmission across any output, then it shall support HDMI audio transmission. Exceptions to this rule for Sources with Type B connectors are found in Appendix B.

An HDMI Source is permitted to transmit L-PCM audio data at sample rates of 32kHz, 44.1kHz, 48kHz, 88.2kHz, 96kHz, 176.4kHz or 192kHz.

An HDMI Source is permitted to transmit IEC 61937 compressed audio at any IEC 60958 frame rate listed in Table 7-4.

If an HDMI Source supports any HDMI audio transmission, then it shall support 2 channel L-PCM (using an IEC 60958 Subpacket structure), with either 32kHz, 44.1kHz or 48kHz sampling rate and a sample size of 16 bits or more.

Transmitted audio shall have an audio sample rate (f_S) within ±1000 ppm of the sample rate indicated in Channel Status bits 24 through 27.

If an HDMI Sink supports audio reception from any input, then it shall support audio reception from all HDMI inputs.

An HDMI Sink may accept L-PCM audio at sample rates of 32kHz, 44.1kHz, 48kHz, 88.2kHz, 96kHz, 176.4kHz or 192kHz, and should indicate these capabilities in the E-EDID data structure.

An HDMI Sink may accept IEC 61937 compressed audio at any IEC 60958 frame rate listed in Table 7-4, and should indicate these capabilities in the E-EDID data structure.

An HDMI Sink that is capable of accepting any audio format is required to accept two channel (IEC 60958-formatted) L-PCM audio at sample rates of 32kHz, 44.1kHz, and 48kHz.

A Sink shall support the reception of an audio stream with correct sample rate indication in Channel Status bits 24 through 27 and with a sample rate (f_S) within ±1000 ppm of any supported sample rate. There is no sample size usage restriction for Sinks.

For CEA-861-D references to Sources, “Basic Audio” is defined as two channel L-PCM audio at sample rates of 32kHz, 44.1kHz, *or* 48kHz, with a sample size of at least 16 bits. For CEA-861-D references to DTV devices, “Basic Audio” is defined as two channel L-PCM audio at sample rates of 32kHz, 44.1kHz, *and* 48kHz.

An HDMI Repeater shall support HDMI audio reception and transmission.

Whenever transmitting a valid audio stream, HDMI Sources shall always include valid and correct information in Channel Status bits 24 through 27. For L-PCM audio, these bits shall indicate the audio sample frequency. For compressed audio formats, these bits shall indicate the IEC 60958 frame rate. An HDMI audio stream shall only indicate values shown in Table 7-4. Note that the allowed values do not include the IEC 60958-specified “Sample frequency not indicated” value.

Table 7-4 Allowed Values for Channel Status bits 24 to 27

Channel Status Bit Number				Sample Frequency or Frame Rate
24	25	26	27	
1	1	0	0	32 kHz
0	0	0	0	44.1 kHz
0	0	0	1	88.2 kHz
0	0	1	1	176.4 kHz
0	1	0	0	48 kHz
0	1	0	1	96 kHz
0	1	1	1	192 kHz
1	0	0	1	768 kHz

Note that rates of 352.8 kHz, 384 kHz, 705.6 kHz are not yet supported by IEC 60958. When this happens, these rates may be supported by an HDMI Source or Sink.

In some cases, pixel-repetition may be required to increase the available bandwidth for audio transmission. For instance, when transmitting a 720x480p video format timing, it is required to pixel double in order to transmit 6 channels @ 96kHz.

7.3.1 One Bit Audio Sample Rate Requirements

A Source may transmit One Bit Audio at an f_S (1/64th of the bit rate) of 32kHz, 44.1kHz, 48kHz, 88.2kHz, 96kHz, 176.4kHz or 192kHz. Any Source capable of supporting One Bit Audio should support an f_S of 44.1kHz, corresponding to a bit rate of 2.8224MHz.

Transmitted One Bit Audio shall have an audio sample rate within ±1000 ppm of the targeted sample rate.

A Sink may accept One Bit Audio at an f_S (1/64th of the bit rate) of 32kHz, 44.1kHz, 48kHz, 88.2kHz, 96kHz, 176.4kHz or 192kHz. Any Sink capable of supporting One Bit Audio shall support an f_S of 44.1kHz, corresponding to a bit rate of 2.8224MHz.

For One Bit Audio, sample frequency information is carried in the Audio InfoFrame (see section 8.2.2).

7.3.2 DST Audio Sample Rate Requirements

All current DST streams carry compressed one bit audio samples with an actual (DAC) sampling frequency of 64*44.1kHz (2.8224MHz).

Any Source capable of supporting DST may transfer DST streams using either DST_Normal or DST_Double rate transmission.

Transmitted DST shall have an audio sample rate within ±1000 ppm of the targeted sample rate.

Any Sink capable of supporting DST shall support streams with a sample rate of 64*44.1kHz (an f_S of 44.1kHz), and shall support reception of both DST_Normal and DST_Double rate.

7.3.3 Video Dependency

Available audio bandwidth depends upon the TMDS clock frequency, the video format timing, and whether or not content protection re-synchronization is needed.

Table 7-5 shows the available audio sample rates for 2-channel (Layout 0) and 8-channel (Layout 1) audio transmission at the various video format timings specified in CEA-861-D, assuming that 58 TMDS clocks of the horizontal blanking interval is required for content protection re-synchronization.

Table 7-5 Max Sampling Frequency for 24-bit Video Format Timings (Informative)*

Description	Format Timing	Pixel Repetition	Vertical Freq (Hz)	Max f_S 8 ch (kHz)	Max frame rate 2 ch, comp**	SuperAudio CD Channel Count
VGA	640x480p	none	59.94/60	48	192	2
480i	1440x480i	2	59.94/60	88.2	192	2
480i	2880x480i	4	59.94/60	192	768	8
240p	1440x240p	2	59.94/60	88.2	192	2
240p	2880x240p	4	59.94/60	192	768	8
480p	720x480p	none	59.94/60	48	192	2
480p	1440x480p	2	59.94/60	176.4	384	8
480p	2880x480p	4	59.94/60	192	768	8
720p	1280x720p	none	59.94/60	192	768	8
1080i	1920x1080i	none	59.94/60	192	768	8
1080p	1920x1080p	none	59.94/60	192	768	8
480i / 120Hz	1440x480i	2	119.9/120	176.4	384	8
480p / 120Hz	720x480p	none	119.9/120	96	384	8
50Hz Formats						
576i	1440x576i	2	50	88.2	192	2
576i	2880x576i	4	50	192	768	8
288p	1440x288p	2	50	88.2	192	2
288p	2880x288p	4	50	192	768	8
576p	720x576p	none	50	48	192	2
576p	1440x576p	2	50	176.4	384	8
576p	2880x576p	4	50	192	768	8

DELL/HP/LENOVO EXHIBIT 1015
Page 478

720p/50	1280x720p	none	50	192	768	8
1080i/50	1920x1080i	none	50	192	768	8
1080p/50	1920x1080p	none	50	192	768	8
1080i, 1250 total	1920x1080i	none	50	192	768	8
576i / 100Hz	1440x576i	2	100	176.4	384	8
576p / 100Hz	720x576p	none	100	96	384	8
1080p @ 24-30Hz						
1080p	1920x1080p	None	24	192	768	8
1080p	1920x1080p	None	25	192	768	8
1080p	1920x1080p	None	29.97/30	192	768	8

** Note that formats listed in Section 6.2.4 but not listed above can carry 8 channels at 192kHz or 8 channels of One Bit Audio at the SuperAudio CD rate.*

*** Note that 2-channel PCM can never exceed 192kHz. Higher values in this column indicate higher frame rates that can be used for compressed streams. Note that 384kHz is not currently supported by IEC 60958.*

7.4 Channel / Speaker Assignment

HDMI allows a Sink to indicate the configuration of attached speakers through the use of the Speaker Allocation Data Block described in CEA-861-D section 7.5.3. Sinks supporting multi-channel L-PCM or multi-channel One Bit Audio shall include this Data Block.

In addition, for L-PCM or One Bit audio streams, the Source shall specify the speaker assignment for each of the channels in the audio stream delivered to the Sink. CEA-861-D section 6.6 specifies the available speaker assignments for active audio channels on HDMI. The indication of the current speaker assignment is carried in the CA field of the Audio InfoFrame.

7.5 Audio, Video Synchronization

For a variety of reasons, an HDMI link may add a delay to the audio and/or video.

An HDMI Source shall be capable of transmitting audio and video data streams with no more than ±2 msec of audio delay relative to the video. Due to the uneven transmission of audio data, the delay shall be considered to be the average delay of all of the audio sample packets over the course of 3 steady-state video frames.

7.6 Audio Data Packetization

Each Subpacket of an Audio Sample Packet shall contain zero or one IEC 60958-defined “frames” of an IEC 60958 or IEC 61937 “block.” There are two defined Subpacket layouts. No others are permitted.

Table 7-6 Audio Packet Layout and Layout Value

Layout Value	Max Num Channels	Samples	Subpkt 0	Subpkt 1	Subpkt 2	Subpkt 3
0	2	4	Chnl 1,2 Sample 0	Chnl 1,2 Sample 1	Chnl 1,2 Sample 2	Chnl 1,2 Sample 3
1	8	1	Chnl 1,2 Sample 0	Chnl 3,4 Sample 0	Chnl 5,6 Sample 0	Chnl 7,8 Sample 0

There are four sample_present bits in the Audio Sample Packet Header, one for each of the Subpackets. These indicate if that Subpacket contains audio sample(s).

In addition, there are four sample_flat.spX bits which are set if no useful audio data was available at the Source during the time period represented by that sample. This may occur during sample rate changes or temporary stream interruptions. When sample_flat.spX is set, Subpacket X continues to represent a sample period but does not contain useful audio data. The sample_flat.spX bit is only valid when the corresponding sample_present.spX bit is set.

Layout 0 can be used to carry up to four samples from a single IEC 61937 or from a single 2-channel IEC 60958 stream of audio.

There are only five valid configurations of sample_present bits for a Layout 0 Audio Packet. They are shown in Table 7-7.

Table 7-7 Valid Sample_Present Bit Configurations for Layout 0

SP0	SP1	SP2	SP3	Description
0	0	0	0	No Subpackets contain audio samples.
1	0	0	0	Only Subpacket 0 contains audio samples.
1	1	0	0	Subpackets 0 and 1 contain audio samples.
1	1	1	0	Subpackets 0, 1, and 2 contain audio samples.
1	1	1	1	All Subpackets contain audio samples.

Layout 1 can be used to carry one audio sample with three to eight channels of L-PCM audio (i.e. two to four IEC 60958 streams).

Valid combinations of sample_present bits for Layout 1 Audio Packets are determined by the permitted channel allocations as described in CEA-861-D section 6.6.

An HDMI Source shall place the data shown into the specified Subpackets and to identify the layout in the Audio Sample Packet Header.

The fields within a Subpacket with a corresponding sample_flat bit set or a sample_present bit clear, are not defined and can be any value.

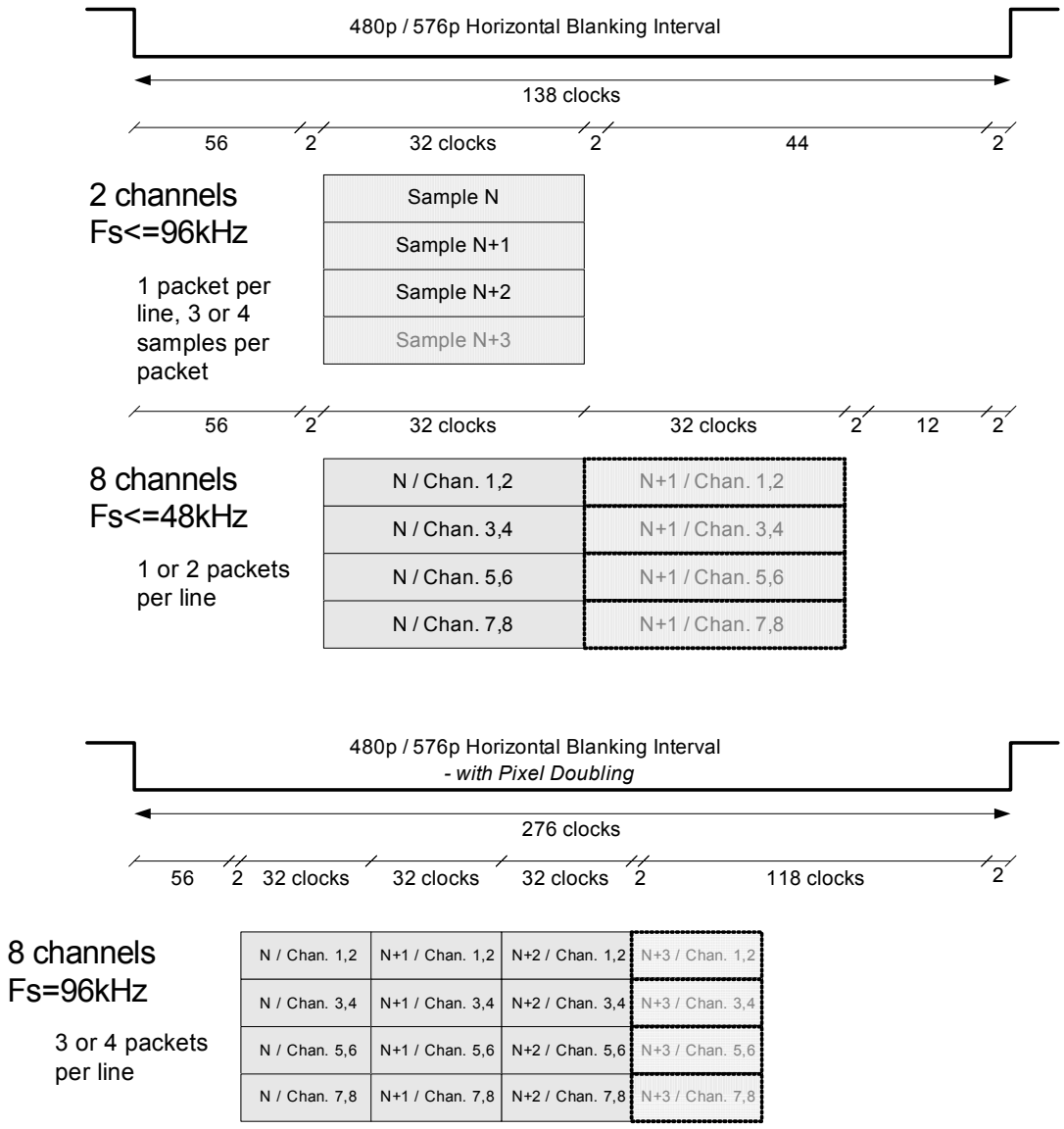


Figure 7-3 Example Audio Sample Timing (Informative)

7.6.1 One Bit Audio Packetization

When transmitting One Bit Audio, each Subpacket shall contain One Bit Audio bits for zero, one or two audio channels.

There are four sample_present bits in the One Bit Audio Sample Packet Header, one for each of the Subpackets. The corresponding bit is set if that Subpacket contains audio samples. There are four samples_invalid.spX bits which are set if no useful audio data was available at the Source during the time period represented by that sample. When samples_invalid.spX is set, Subpacket X continues to represent a sample period but does not contain any useful data.

Layout 0 can be used to carry 2 channels of One Bit Audio samples. Layout 1 can be used to carry from three to eight channels of One Bit Audio samples.

Valid combinations of sample_present bits for Layout 1 Audio Packets are determined by the permitted channel allocations as described in section 7.6 above.

The fields within a Subpacket with a corresponding samples_invalid bit set or a sample_present bit clear, are not defined and can be any value.

7.6.2 High-Bitrate Audio Stream Packetization

When carrying IEC 61937 compressed audio at frame rates above 192kHz, the HBR Audio Stream packet shall be used.

For many high bitrate streams (e.g. DTS-HD Master Audio and Dolby MAT), the IEC 61937 data bursts will always have a repetition period that is a multiple of four frames, and so the Pa and Pb syncwords will always be found in the same subpacket. In such cases, the codec vendor may impose the additional constraint that Pa and Pb always appear in subpacket 0.

7.6.3 DST Packetization

The DST audio stream consists of a series of frames, each of which carries audio data for 1/75th of a second. Each DST frame will be transmitted using a number of DST Audio Packets. The end of a DST frame may occur within a DST Audio Packet. Any unused bits in such a DST Audio Packet shall be padded with “0”. In the time between the completion of one DST Audio Frame and the start of the next, DST Audio Packets containing all “0” shall be sent. A new DST Audio Packet with the frame_start bit set to “1” in the header shall be used when the next DST Audio Frame starts and the first bit of the new DST frame data shall be placed in D.0 of the packet body.

When samples_invalid in the DST Audio Packet is set to “1”, then the data in the DST Audio Packet is not valid or does not contain any useful data.

7.7 Error Handling (Informative)

The behavior of the Sink after detecting an error is implementation-dependent. However, Sinks should be designed to prevent loud spurious noises from being generated due to errors. Sample repetition and interpolation are well known concealment techniques and are recommended.

7.8 Packet Delivery Rules

7.8.1 Audio Sample Packets

All audio samples that are stored in a source buffer shall be sent as soon as possible while still fulfilling requirements for audio/video synchronization, and Data Island timing and placement. When using Layout 0 Audio Sample Packets, the Source shall transmit an Audio Sample Packet if at least one sample is stored in the source buffer.

Relative to an ideal constant-frequency clock, the jitter present in the Audio Sample Packet transmission timing shall not exceed one horizontal line period plus a single audio sample period.

7.8.2 Audio Clock Regeneration Packets

Nominally, Audio Clock Regeneration Packets with newly generated CTS values will be transmitted at a rate of $128*f_S/N$. On average, the Source shall transmit CTS values at this rate precisely. The Source shall transmit each CTS data value as close as possible to the nominal transmission time for that value with the exception that priority must be given to Audio Sample packets to ensure that Audio Sample Packet delivery requirements are met.

7.9 One Bit Audio Usage Overview

One Bit Audio data is transmitted using the One Bit Audio packet defined in section 5.3.9 and described in section 7.6.1.

One Bit Audio clock regeneration uses the same mechanism used for all audio on HDMI and is described in section 7.2.4. One Bit Audio sample rate requirements are described in section 7.3.1.

A Sink may indicate its support for One Bit Audio with the Short Audio Descriptor as described in section **Error! Reference source not found.**.

One Bit Audio uses some fields within the Audio InfoFrame differently than L-PCM or compressed audio; these differences are described in section 8.2.2.

7.10 DST Usage Overview

DST data is transmitted using the DST Sample packet defined in section 5.3.10 and described in section 7.6.3.

DST clock regeneration uses the same mechanism used for all audio on HDMI and is described in section {7.2.5}. One Bit Audio sample rate requirements are described in section 7.3.2.

A Sink may indicate its support for DST with the Short Audio Descriptor as described in CEA-861-D.

In some cases, DST uses some fields within the Audio InfoFrame differently than L-PCM or IEC 61937 compressed audio; these differences are described in section 8.2.2.

8 Control And Configuration

8.1 Overview

HDMI includes three separate communications channels: TMDS, DDC, and the optional CEC.

TMDS is used to carry all audio and video data as well as auxiliary data, including AVI and Audio InfoFrames that describe the active audio and video streams.

The DDC channel is used by an HDMI Source to determine the capabilities and characteristics of the Sink by reading the E-EDID data structure.

HDMI Sources are expected to read the Sink's E-EDID and to deliver only the audio and video formats that are supported by the Sink. In addition, HDMI Sinks are expected to detect InfoFrames and to process the received audio and video data appropriately.

The CEC channel is optionally used for higher-level user functions such as automatic setup tasks or tasks typically associated with infrared remote control usage.

8.2 InfoFrames

An InfoFrame packet carries one InfoFrame. The InfoFrame provided by HDMI is limited to 30 bytes plus a checksum byte. HDMI Sources are required, in some cases, to use the AVI InfoFrame and Audio InfoFrame and recommended in other cases. Other InfoFrames specified in CEA-861-D are optional.

All InfoFrames are described in detail in CEA-861-D. The following describes how two of these InfoFrames are placed within the InfoFrame Packet structure and any areas where HDMI behavior is different from that specified in CEA-861-D.

8.2.1 Auxiliary Video information (AVI) InfoFrame

Various aspects of the current video stream are indicated by the HDMI Source to the Sink with an Auxiliary Video information (AVI) InfoFrame.

A Source shall always transmit an AVI InfoFrame at least once per two video fields if the Source:

- is *ever* capable of transmitting an AVI InfoFrame or,
- is ever capable of transmitting YC_BC_R pixel encoding or,
- is ever capable of transmitting any colorimetry other than the transmitted video format's default colorimetry or,
- is ever capable of transmitting any xvYCC or future enhanced colorimetry or,
- is ever capable of transmitting any Gamut Metadata packet or,
- is ever capable of transmitting any video format with multiple allowed pixel repetitions.

The AVI InfoFrame shall be transmitted even while such a Source is transmitting RGB and non-pixel-repeated video. When a Source is not explicitly required to transmit AVI InfoFrames, it is recommended that the Source transmit AVI InfoFrames.

The packetization of the AVI InfoFrame Version 2 is shown below.

Table 8-1 AVI InfoFrame Packet Header

Byte \ Bit #	7	6	5	4	3	2	1	0
HB0	Packet Type = 0x82							
HB1	Version = 0x02							
HB2	0	0	0	Length = 13 (0x0D)				

Table 8-2 AVI InfoFrame Packet Contents

Packet Byte #	CEA-861-D Byte #	7	6	5	4	3	2	1	0
PB0	N. A.	Checksum							
PB1	Data Byte 1	Rsvd (0)	Y1	Y0	A0	B1	B0	S1	S0
PB2	Data Byte 2	C1	C0	M1	M0	R3	R2	R1	R0
PB3	Data Byte 3	ITC	EC2	EC1	EC0	Q1	Q0	SC1	SC0
PB4	Data Byte 4	Rsvd (0)	VIC6	VIC5	VIC4	VIC3	VIC2	VIC1	VIC0
PB5	Data Byte 5	Reserved (0)				PR3	PR2	PR1	PR0
PB6	Data Byte 6	Line Number of End of Top Bar (lower 8 bits)							
PB7	Data Byte 7	Line Number of End of Top Bar (upper 8 bits)							
PB8	Data Byte 8	Line Number of Start of Bottom Bar (lower 8 bits)							
PB9	Data Byte 9	Line Number of Start of Bottom Bar (upper 8 bits)							
PB10	Data Byte 10	Pixel Number of End of Left Bar (lower 8 bits)							
PB11	Data Byte 11	Pixel Number of End of Left Bar (upper 8 bits)							
PB12	Data Byte 12	Pixel Number of Start of Right Bar (lower 8 bits)							
PB13	Data Byte 13	Pixel Number of Start of Right Bar (upper 8 bits)							
PB14-PB27	n. a.	Reserved (0)							

See CEA-861-D section 6.4 for more information on the following fields:

- Y0, Y1 RGB or YC_BC_R indicator.
- A0 Active Information Present. Indicates whether field R0...R3 is valid. See CEA-861-D table 8 for details.
- B0, B1 Bar Info data valid. See CEA-861-D table 8 for details.

- S0, S1 Scan Information (i.e. overscan, underscan). See CEA-861-D table 8 for details.
- C0, C1 Colorimetry (ITU BT.601, BT.709 etc.). See CEA-861-D table 9 for details.
- EC0, EC1, EC2 Extended Colorimetry (IEC 61966-2-4 etc.). See CEA-861-D table 11 for details.
- Q1, Q0 Quantization range (Full vs. Limited, etc.). See CEA-861-D table 11 for details.
- ITC IT Content. See CEA-861-D table 11 for details.
- M0, M1 Picture Aspect Ratio (4:3, 16:9). See CEA-861-D table 9 for details.
- R0...R3 Active Format Aspect Ratio. See CEA-861-D table 10 and Annex H for details.
- VIC0...VIC6 Video Format Identification Code. When transmitting any video format in section 6.2.4, above, an HDMI Source shall set the VIC field to the Video Code for that format. See CEA-861-D section 6.4 for details.
- PR0...PR3 Pixel Repetition factor. See CEA-861-D table 13 for details.
- SC1, SC0 Non-uniform Picture Scaling. See CEA-861-D table 11.

Table 8-3 HDMI Valid Pixel Repeat Values for Each Video Format Timing

Video Code	Video Description		HDMI Pixel Repeat Values
1	640x480p @ 60Hz		No Repetition
2,3	720x480p @ 59.94/60Hz		No Repetition
4	1280x720p @ 59.94/60Hz		No Repetition
5	1920x1080i @ 59.94/60Hz		No Repetition
6,7	720(1440)x480i @ 59.94/60Hz		Pixel sent 2 times
8,9	720(1440)x240p @ 59.94/60Hz		Pixel sent 2 times
10,11	2880x480i @ 59.94/60Hz		Pixel sent 1 to 10 times
12,13	2880x240p @ 59.94/60Hz		Pixel sent 1 to 10 times
14,15	1440x480p @ 59.94/60Hz		Pixel sent 1 to 2 times
16	1920x1080p @ 59.94/60Hz		No Repetition
17,18	720x576p @ 50Hz		No Repetition
19	1280x720p @ 50Hz		No Repetition
20	1920x1080i @ 50Hz		No Repetition
21,22	720(1440)x576i @ 50Hz		Pixel sent 2 times
23,24	720(1440)x288p @ 50Hz		Pixel sent 2 times
25,26	2880x576i @ 50Hz		Pixel sent 1 to 10 times
27,28	2880x288 @ 50Hz		Pixel sent 1 to 10 times
29,30	1440x576p @ 50Hz		Pixel sent 1 to 2 times
31	1920x1080p @ 50Hz		No Repetition
32	1920x1080p @ 23.97/24Hz		No Repetition
33	1920x1080p @ 25Hz		No Repetition
34	1920x1080p @ 29.97/30Hz		No Repetition
35,36	2880x480p @ 59.94/60Hz		Pixel sent 1, 2 or 4 times
37,38	2880x576p @ 50Hz		Pixel sent 1, 2 or 4 times
39	1920x1080i (1250 total) @ 50Hz		No Repetition
40	1920x1080i @ 100Hz		No Repetition
41	1280x720p @ 100Hz		No Repetition
42,43	720x576p @ 100Hz		No Repetition
44,45	720(1440)x576i @ 100Hz		Pixel sent 2 times
46	1920x1080i @ 119.88/120Hz		No Repetition
47	1280x720p @ 119.88/120Hz		No Repetition
48,49	720x480p @ 119.88/120Hz		No Repetition
50,51	720(1440)x480i @ 119.88/120Hz		Pixel sent 2 times
52,53	720X576p @ 200Hz		No Repetition
54,55	720(1440)x576i @ 200Hz		Pixel sent 2 times
56,57	720x480p @ 239.76/240Hz		No Repetition
58,59	720(1440)x480i @ 239.76/240Hz		Pixel sent 2 times

8.2.2 Audio InfoFrame

A Source shall indicate characteristics of the active audio stream using the IEC 60958 Channel Status bits, IEC 61937 Burst Info and/or stream data (if present) and the Audio InfoFrame. Whenever an active audio stream is being transmitted, an accurate Audio InfoFrame shall be transmitted at least once per two video fields.

Upon the start of a new audio stream or upon any change in the audio stream that can be indicated by the Audio InfoFrame, a modified, accurate Audio InfoFrame shall be transmitted no later than one video field following the first affected non-silent audio sample. Preferably, this

would occur just before the first affected audio sample is transmitted. For One Bit Audio streams, the Audio InfoFrame shall be transmitted before the first affected sample.

The Audio InfoFrame transmission may occur at any time that a Data Island packet may be transmitted, including during any horizontal or vertical blanking period.

Note that several of the fields permit a value of 0 (referred to in the CEA-861-D specification as “Refer to Stream Header”). A value of 0 signifies that the information associated with that field is actually indicated or implied by other items in the audio stream, for instance, by the IEC 60958 Channel Status bits or the IEC 61937 Burst Info.

The following tables show the packetization of the Audio InfoFrame.

Table 8-4 Audio InfoFrame Packet Header

Byte \ Bit #	7	6	5	4	3	2	1	0
HB0	Packet Type = 0x84							
HB1	Version Number = 0x01							
HB2	0	0	0	Length = 10 (0x0A)				

Table 8-5 Audio InfoFrame Packet contents

Packet Byte #	CEA-861-D Byte #	7	6	5	4	3	2	1	0
PB0	n. a.	Checksum							
PB1	Data Byte 1	CT3	CT2	CT1	CT0	Rsvd	CC2	CC1	CC0
PB2	Data Byte 2	Reserved (0)			SF2	SF1	SF0	SS1	SS0
PB3	Data Byte 3	Format depends on coding type (i.e. CT0...CT3)							
PB4	Data Byte 4	CA7	CA6	CA5	CA4	CA3	CA2	CA1	CA0
PB5	Data Byte 5	DM_INH	LSV3	LSV2	LSV1	LSV0	Reserved (0)		
PB6	Data Byte 6	Reserved (0)							
PB7	Data Byte 7	Reserved (0)							
PB8	Data Byte 8	Reserved (0)							
PB9	Data Byte 9	Reserved (0)							
PB10	Data Byte 10	Reserved (0)							
PB11-PB27	n. a.	Reserved (0)							

See CEA-861-D section 6.6 for more information on the following fields:

- CC0...CC2 Channel Count. See CEA-861-D table 17 for details.
- CT0...CT3 Coding Type. The CT bits shall always be set to a value of 0 (“Refer to Stream Header”).
- SS0...SS1 Sample Size. The SS bits shall always be set to a value of 0 (“Refer to Stream Header”).
- SF0...SF2 Sample Frequency. See CEA-861-D table 18 for details. For L-PCM and IEC 61937 compressed audio streams, the SF bits shall always be set to a value of 0 (“Refer to Stream Header”). For One Bit Audio and DST streams, the value indicated by the SF bits shall equal the ACR f_S value (see sections 7.2.5 and 7.2.6). For Super Audio CD, the SF bits are typically set to 0, 1, 0, to indicate a Sample Frequency of 2.8224MSamples/s (i.e. 64*44.1kHz).
- CA0...CA7 Channel/Speaker Allocation. See CEA-861-D Section 6.6.2 for details. The CA field is not valid for IEC 61937 compressed audio streams.
- LSV0...LSV3 Level Shift Value (for downmixing). See CEA-861-D Section 6.6.2 and CEA-861-D table 21 for details.
- DM_INH Downmix Inhibit. See CEA-861-D section 6.6.2 and table 22 for details. The DM_INH field is to be set only for DVD-Audio applications and corresponds to the value in the DM_INH field of the current audio stream

being played from the disk. The DM_INH field value shall be set to zero in all cases other than DVD-Audio applications.

Data Byte 3 shall always be set to a value of 0.

8.3 E-EDID Data Structure

All Sinks shall contain an CEA-861-D compliant E-EDID data structure accessible through the DDC.

A Source shall read the EDID 1.3 and first CEA Extension to determine the capabilities supported by the Sink. Additional extensions may be read to discover additional capabilities. The Source is responsible for any format conversions that may be necessary to supply audio and video in an understandable form to the Sink. However, it is permitted for a Source to transmit Basic Audio (see Section 7.2.6) to a Sink that does not indicate support for Basic Audio.

The Source shall not transmit at TMDS clock rates higher than the maximum rate supported by the Sink, as determined by video format and Deep Color mode support indications but limited by the Max_TMDS_Clock field of the HDMI VSDB.

The overall structure of the E-EDID in the Sink shall conform to the E-EDID structure defined in the VESA E-EDID Standard Release A, Revision 1, but shall also meet the additional requirements specified herein.

The first 128 bytes of the E-EDID shall contain an EDID 1.3 structure. The contents of this structure shall also meet the requirements of CEA-861-D.

8.3.1 CEA Extension

The first E-EDID 'extension' shall contain a CEA Extension version 3, defined in CEA-861-D section 7.5. Additional CEA Extensions may also be present. The E-EDID shall not contain a CEA Extension version 1 or version 2.

CEA Extension version 3 details are described in CEA-861-D Section 7.5.

Further details on the requirements of the data structures in the E-EDID and implementation examples are given in CEA-861-D.

8.3.2 HDMI Vendor-Specific Data Block (HDMI VSDB)

The first CEA Extension shall include an HDMI Vendor Specific Data Block (HDMI VSDB) shown in Table 8-6. This is a CEA-861-D Vendor Specific Data Block (see CEA-861-D section 7.5.4 for details) containing a 24-bit IEEE Registration Identifier of 0x000C03, a value belonging to HDMI Licensing, LLC.

Sinks shall contain an HDMI VSDB minimally containing a 2-byte Source Physical Address field following the 24-bit identifier. An HDMI VSDB may have zero or more extension fields as shown in Table 8-6. The minimum value of N (length) is 5 and the maximum value of N is 31. A Sink that supports any function indicated by an extension field shall use an HDMI VSDB with a length sufficient to cover all supported fields.

The Source shall have the ability to handle an HDMI VSDB of any length. In future specifications, new fields may be defined. These additional fields will be defined such that a zero value indicates the same characteristics as is indicated if the field was not present. Sources should use the

length field to determine which extension fields are present, and shall process the HDMI VSDB with no regard to non-zero values in fields defined as Reserved in this specification.

Table 8-6 HDMI-LLC Vendor-Specific Data Block (HDMI VSDB)

Byte #	7	6	5	4	3	2	1	0
0	Vendor-specific tag code (=3)				Length (=N)			
1	24-bit IEEE Registration Identifier (0x000C03) (least significant byte first)							
2								
3								
4	A				B			
5	C				D			
6	Supports_AI	DC_48bit	DC_36bit	DC_30bit	DC_Y444	Rsvd (0)	Rsvd (0)	DVI_Dual
7	Max_TMDS_Clock							
8	Latency_Fields_Present	I_Latency_Fields_Present	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)	Rsvd (0)
(9)	Video_Latency							
(10)	Audio_Latency							
(11)	Interlaced_Video_Latency							
(12)	Interlaced_Audio_Latency							
9, 11 or 13...N*	*Reserved (0)***							

extension fields ↓

* *The position of these bytes will depend upon the values of Latency_Fields_Present and I_Latency_Fields_Present.*

** *No additional bytes are necessary but if present, they shall be zero.*

- A, B, C, D — [4 bits each] Components of Source Physical Address (A.B.C.D). See Section 8.7.
- Supports_AI — [1 bit] Set to 1 if the Sink supports at least one function that uses information carried by the ACP, ISRC1, or ISRC2 packets. If Supports_AI is set (=1), then the Sink shall accept and process any ACP, ISRC1 or ISRC2 packet with no regard to non-zero values in fields defined as Reserved in this specification. If the Sink does not support ACP, ISRC1 or ISRC2 packets, Supports_AI shall be clear (=0).
- DC_30bit — [1 bit] Set if Sink supports 30 bits/pixel (10 bits/color).
- DC_36bit — [1 bit] Set if Sink supports 36 bits/pixel (12 bits/color).

- DC_48bit [1 bit] Set if Sink supports 48 bits/pixel (16 bits/color).
- DC_Y444 [1 bit] Set if Sink supports YC_BC_R 4:4:4 in Deep Color modes.

The three DC_XXbit bits above only indicate support for RGB 4:4:4 at that pixel size. Support for YC_BC_R 4:4:4 in Deep Color modes is indicated with the DC_Y444 bit. If DC_Y444 is set, then YC_BC_R 4:4:4 is supported for all modes indicated by the DC_XXbit flags. This provides the Sink the flexibility of supporting YC_BC_R formats for the standard color-depth (24-bits/pixel) while only supporting RGB for Deep Color modes.

- DVI_Dual [1 bit] Set if Sink supports DVI dual-link operation.
- Max_TMDS_Clock [1 byte] Indicates the maximum TMDS clock rate supported. Max rate = Max_TMDS_Clock * 5MHz. This field shall be set correctly and non-zero if the Sink supports TMDS clock frequencies above 165MHz or supports any Deep Color mode or supports DVI dual-link. A value of zero means that no clock rate is indicated.

The Max_TMDS_Clock field may be set by the Sink at a level below the TMDS clock rate corresponding to the maximum pixel clock rate at the maximum color depth. This allows the Sink to support higher color depths at lower resolutions than it can support at higher resolutions.

See section 8.9 for more detail on the following lipsync-related fields:

- Latency_Fields_Present [1 bit] If set (=1) then the Video_Latency and Audio_Latency fields are present. If clear (=0) then these fields are not present in the HDMI VSDB.
- I_Latency_Fields_Present [1 bit] If set (=1) then the latency fields total four bytes, two for video and audio latency information when progressive video formats are received and two for latency information when interlaced video formats are received. If clear (=0) then only two bytes are present, indicating the video and audio latency when any video format is received. I_Latency_Fields_Present shall be zero if Latency_Fields_Present is zero.
- Video_Latency [1 byte] Indicates the amount of video latency when receiving any video format or only when receiving progressive video formats; if I_Latency_Fields_Present flag == 1 then this field only indicates the latency while receiving progressive video formats, otherwise this field indicates the latency when receiving any video format. Value is number of (milliseconds / 2) + 1 with a maximum allowed value of 251 (indicating 500 millisecond duration). A value of 0 indicates that the field is not valid or that the latency is unknown. A value of 255 indicates that no video is supported in this device or downstream.
- Audio_Latency [1 byte] Indicates the amount of audio latency when receiving any video format or only when receiving progressive video formats; if I_Latency_Fields_Present flag == 1 then this field only indicates the latency while receiving progressive video formats, otherwise this field indicates the latency when receiving any video format. Value is number of (milliseconds / 2) + 1 with a maximum allowed value of 251 (indicating 500 millisecond duration). A value of 0 indicates that the field is not valid or that the latency is unknown. A value of 255 indicates that no audio is supported in this device or downstream.
- Interlaced_Video_Latency [1 byte] This field is only present if I_Latency_Fields_Present flag == 1. If present, the field indicates the amount of video latency when

receiving an interlaced video format. Format is identical to Video_Latency field.

- Interlaced_Audio_Latency [1 byte] This field is only present if I_Latency_Fields_Present flag == 1. If present, the field indicates the amount of audio latency when receiving an interlaced video format. Format is identical to Audio_Latency field.

8.3.3 DVI/HDMI Device Discrimination

In order to determine if a sink is an HDMI device, an HDMI Source shall check the E-EDID for the presence of an HDMI Vendor Specific Data Block within the first CEA Extension. Any device with an HDMI VSDB of any valid length, containing the IEEE Registration Identifier of 0x000C03, shall be treated as an HDMI device.

Any device with an E-EDID that does not contain a CEA Extension or does not contain an HDMI VSDB of any valid length shall be treated by the Source as a DVI device (see Appendix C).

8.3.4 Audio and Video Details

Sink audio characteristics and support are indicated in a series of Short Audio Descriptors located in the CEA Extension's Data Block collection. This data includes a list of audio encodings supported by the Sink and parameters associated with each of those encodings, such as number of channels supported for that format. A Speaker Allocation Descriptor may also be included in the Data Block collection and is required for Sinks supporting multi-channel L-PCM or multi-channel One Bit Audio.

A Sink may indicate support for YC_BC_R pixel encodings. To indicate support, bits 4 and 5 of byte 3 of the CEA Extension shall both be set to one (see Table 27 of CEA-861-D). To indicate no support, bits 4 and 5 shall both be zero.

With the exception of 640x480p video format, if a Sink is required to support a particular video format, video format timing, or pixel encoding, then the Sink shall indicate support for that video format, video format timing or pixel encoding in the E-EDID. Explicit indication of 640x480p is optional but is not required because all Sinks are required to support that video format.

To indicate support for any video format in section 6.2.4, an HDMI Sink shall use a Short Video Descriptor (SVD) containing the Video Code for that format and may also use a Detailed Timing Descriptor (DTD).

If the Sink supports extended colorimetries (those beyond the default standard- and high-definition colorimetries) or supports the reception of gamut-related metadata, the Sink shall use a Colorimetry Data Block to indicate support for these colorimetries and metadata. See section 6.7.3 for more details.

8.4 Enhanced DDC

Enhanced DDC described in this section is defined in VESA "ENHANCED DISPLAY DATA CHANNEL STANDARD Version 1 (September 2, 1999)". All Sinks are required to support these

enhanced DDC features. If a Sink's E-EDID structure is longer than 256 bytes, it shall support the segment pointer.

8.4.1 Timing

Data is synchronized with the SCL signal and timing shall comply with the Standard Mode of the I^2C specification (100kHz maximum clock rate).

I^2C Bus is a standard two-wire (clock and data) serial data bus protocol. Refer to the I^2C specification for details.

Note that an HDMI Sink may hold off the DDC transaction by stretching the SCL line during the SCL-low period following the Acknowledge bit as permitted by the I^2C specification. All HDMI Sources shall delay the DDC transaction while the SCL line is being held low.

8.4.2 Data Transfer Protocols

The Source shall use I^2C commands to read information from a Sink's E-EDID with a slave address.

In Enhanced DDC, a segment pointer is used to allow addressing of the E-EDID outside of the normal 256-byte limit of the 0xA0/0xA1 address. The Enhanced DDC protocol sets the segment pointer before the remainder of the DDC command.

8.4.3 Segment pointer

Enhanced DDC allows access of up to 32 Kbytes of data. This is accomplished using a combination of the 0xA0/0xA1 address pair and a segment pointer. For each value of the segment pointer, 256 bytes of data are available at the 0xA0/0xA1 address pair. An unspecified segment pointer references the same data as when the segment pointer is zero.

Each successive value of the segment pointer allows access to the next two blocks of E-EDID (128 bytes each). The value of the segment pointer register cannot be read since it is reset at the completion of each command.

8.4.4 Enhanced DDC Sink

The Sink shall be Enhanced DDC read compliant.

The Sink shall be capable of responding with EDID 1.3 data and up to 255 extension blocks, each 128 bytes long (up to 32K bytes total E-EDID memory) whenever the Hot Plug Detect signal is asserted.

The Sink should be capable of providing E-EDID information over the Enhanced DDC channel whenever the +5V Power signal is provided. This should be available within 20msec after the +5V Power signal is provided.

8.4.5 Enhanced DDC Source

The Source shall use Enhanced DDC protocols.

The Source shall be capable of reading EDID 1.3 data at DDC address 0xA0.

The Source reads Enhanced EDID extensions data at DDC address 0xA0 using segment pointer 0x60.

8.5 Hot Plug Detect Signal

An HDMI Sink shall not assert high voltage level on its Hot Plug Detect pin when the E-EDID is not available for reading. This requirement shall be fulfilled at all times, even if the Sink is powered-off or in standby. The Hot Plug Detect pin may be asserted only when the +5V Power line from the Source is detected. This will ensure that the Hot Plug Detect pin is not asserted before the Third Make of the connector (see Section 0).

A Source may use a high voltage level Hot Plug Detect signal to initiate the reading of E-EDID data.

A Source shall assume that any voltage within the range specified for High voltage level in Table 4-26 indicates that a Sink is connected and that E-EDID is readable. It does not indicate whether or not the Sink is powered or whether or not the HDMI input on the Sink is selected or active.

An HDMI Sink shall indicate any change to the contents of the E-EDID by driving a low voltage level pulse on the Hot Plug Detect pin. This pulse shall be at least 100 msec.

8.6 Consumer Electronics Control (CEC)

The CEC line is used for high-level user control of HDMI-connected devices. The mandatory requirements for the CEC line are described in detail in Section 4.2.10, CEC Line. The optional CEC protocol is described in Supplement 1: Consumer Electronics Control (CEC).

8.7 Physical Address

8.7.1 Overview

In order to allow CEC to be able to address specific physical devices and control switches, all devices shall have a physical address. This connectivity has to be worked out whenever a new device is added to the cluster. The physical address discovery process uses only the DDC/EDID mechanism and applies to all HDMI Sinks and Repeaters, not only to CEC-capable devices.

The CEC and DDC connections are shown in Figure 8-1.

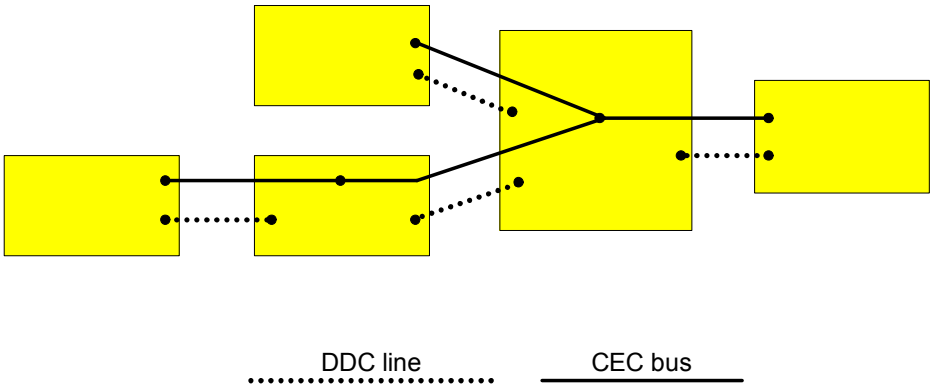


Figure 8-1 CEC and DDC line connections

The CEC line is directly connected to all nodes on the network.

After discovering their own physical address, the CEC devices transmit their physical and logical addresses to all other devices, thus allowing any device to create a map of the network.

8.7.2 Physical Address Discovery

The physical address of each node is determined through the physical address discovery process. This process is dynamic in that it automatically adjusts physical addresses as required as devices are physically or electrically added or removed from the device tree.

All Sinks and Repeaters shall perform the steps of physical address discovery and propagation even if those devices are not CEC-capable. Sources are not required to determine their own physical address unless they are CEC-capable.

All addresses are 4 digits long allowing for a 5–device-deep hierarchy. All are identified in the form of n.n.n.n in the following description. An example of this is given in Figure 8-3.

A Sink or a Repeater that is acting as the CEC root device will generate its own physical address: 0.0.0.0. A Source or a Repeater reads its physical address from the EDID of the connected Sink. The CEC line may be connected to only one HDMI output so a device with multiple HDMI outputs will read its physical address from the EDID on the CEC-connected output. Each Sink and Repeater is responsible for generating the physical address of all Source devices connected to that device by appending a port number onto its own physical address and placing that value in the EDID for that port. The Source Address Field of the HDMI Vendor Specific Data Block (see Section 8.3.2) is used for this purpose.

Note that the values shown in the figures below represent the physical addresses for the devices themselves, not the Source physical addresses stored in the EDID within that device. In fact, for all devices shown, except the TV, those physical addresses are stored in the EDID of the connected Sink. An example is shown for the TV at physical address 0.0.0.0.

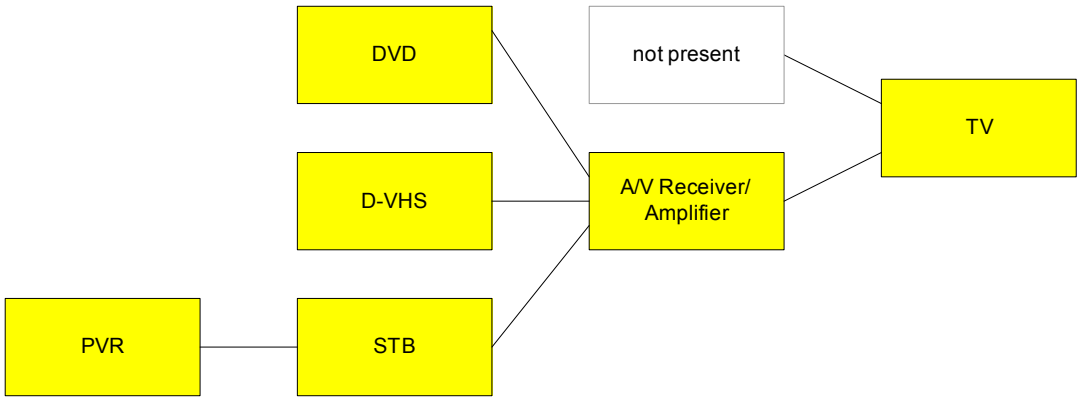


Figure 8-2 Typical HDMI cluster

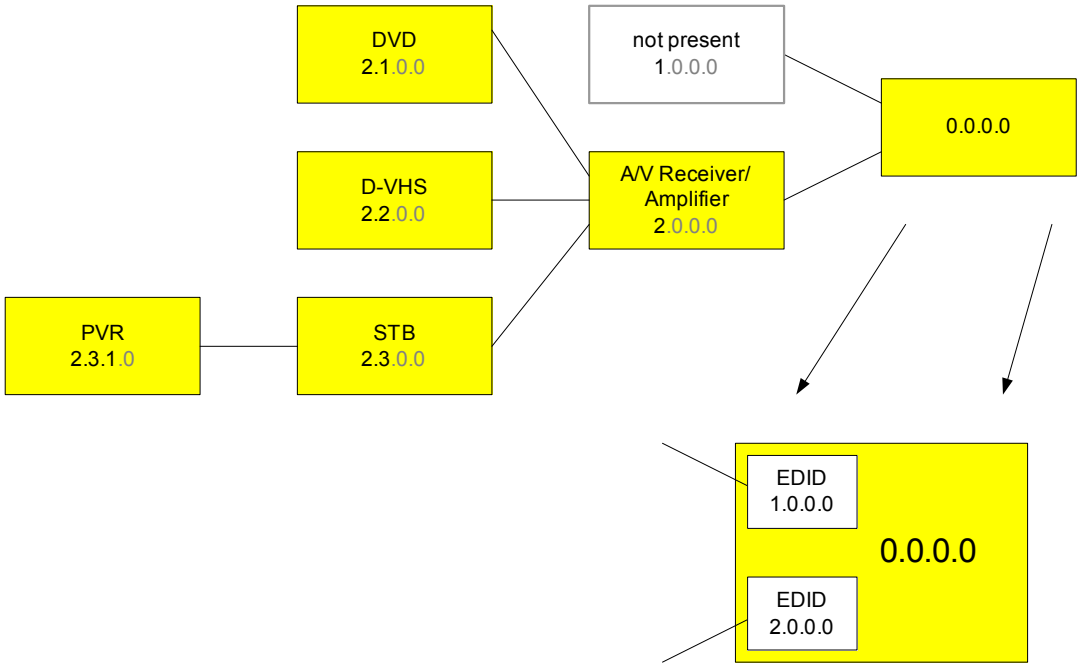


Figure 8-3 Addresses within an HDMI cluster

8.7.3 Discovery Algorithm

The following algorithm is used to allocate the physical address of each device whenever HPD is de-asserted or upon power-up:

```
Disable assertion of HPD to all source devices
If I am CEC root
    Set my_address to 0.0.0.0
Else
    Wait for HPD from sink
    Query sink for my_address of my connection (Section 8.7.4)
    The device shall retain this physical address until HPD is
        removed (or the device is powered off).
End if
If device has connections for source devices then
    Label all possible connections to source devices uniquely starting
        from connection_label = 1 to the number of source input connections
    If device has separate EDIDs for each source connection then
        If my_address ends with 0 then
            Set each source_physical_address to my_address with the
                first 0 being replaced with connection_label.
        Else (i.e. beyond the fifth layer of the tree)
            Set each source_physical_address to F.F.F.F
        End if
    Else
        Set each source_physical_address to my_address
    End if
    Write source_physical_address to HDMI VSDB in EDID for each source
        connection
End if
Allow HPD to be asserted for source devices
```

8.7.4 HDMI Sink Query

A Source shall determine its physical address (my_address) by checking the HDMI Vendor Specific Data Block (see Section 8.3.2) within the EDID. The fourth and fifth bytes of this 5 byte structure contain the Source Physical Address (fields A, B, C, D).

8.8 ISRC Handling

A Source shall not transmit an ISRC1 or ISRC2 Packet to a Sink that does not have Supports_AI = 1.

A Source may handle an International Standard Recording Code (ISRC) and/or UPC/EAN describing the origin or owner details for each track of content on the medium. These values may be transmitted using the ISRC1 and ISRC2 packets.

When fields UPC_EAN_ISRC_16 through 31 include effective data (i.e. not "reserved"), a subsequent ISRC2 Packet shall be transmitted. In other cases, the ISRC2 packet may optionally be transmitted.

When a subsequent ISRC2 Packet is transmitted, the ISRC_Cont field shall be set and shall be clear otherwise.

For further description of the UPC_EAN_ISRC fields, see "DVD Specifications for Read-Only Disc", Part 4: AUDIO SPECIFICATIONS Version 1.0, March 1999, Annex B".

Regarding usage of the ISRC_Status field, Source shall comply with "DVD Specifications for Read-Only Disc", "Part 4: AUDIO SPECIFICATIONS", Version-up Information (from 1.1 to 1.2), Table 7.2.3.1.1-2, May 2000. Following is a summary of the relevant rules from that specification:

- At the beginning of each track, at least two complete UPC_EAN_ISRC codes are transmitted with an ISRC_Status of 0b001.
- During the bulk of the track, continuous repetitions of the packet(s) are required, with an ISRC_Status of 0b010.
- Immediately before the end of each track, at least two complete UPC_EAN_ISRC codes are transmitted with an ISRC_Status of 0b100.

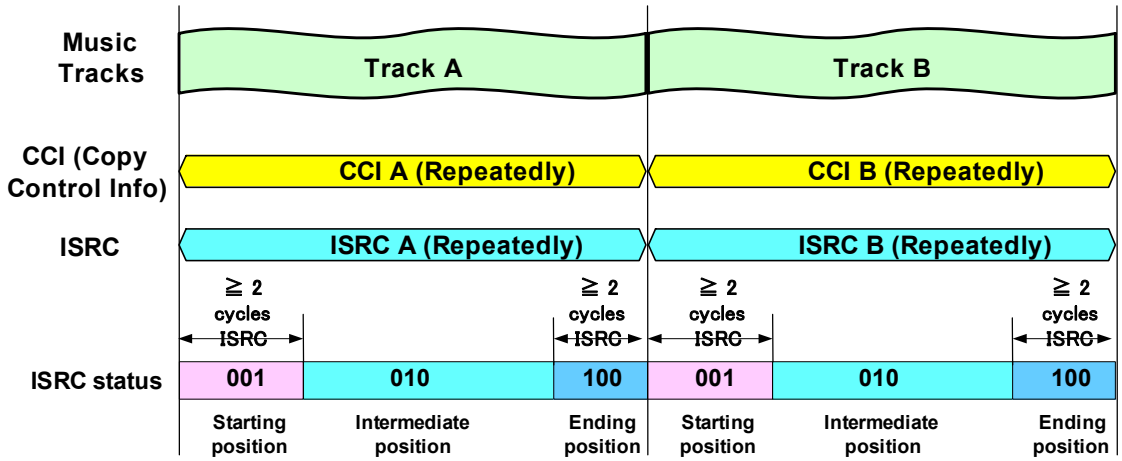


Figure 8-4 ISRC/CCI and ISRC Status Handling

8.9 Auto Lipsync Correction Feature

Some common home theater device configurations will render the audio in a device other than the TV. In these configurations, the video processing latency of the TV may cause perceptible lipsync issues to the user. These issues can be corrected by delaying the audio to compensate for the video processing latency. The HDMI Auto Lipsync Correction feature allows a Source or Repeater to automatically determine the necessary amount of audio delay before presentation or output of that audio signal.

8.9.1 EDID Latency Info

HDMI Sinks and Repeaters may declare audio and video latency information in the EDID, allowing an HDMI Source or Repeater to determine how best to maintain synchronization between the rendered audio and video. These fields and other lipsync-related fields are in the HDMI VSDB (see section 8.3.2). The latency values within these fields indicate the amount of time between the video or audio entering the HDMI input to the actual presentation to the user (on a display or speakers), whether that presentation is performed by the device itself or by another device downstream.

Many TVs internally compensate for their own video processing latency by adding a delay to the audio stream that corresponds to the video latency. In this case, the EDID-indicated audio and video latencies will be equal. This delay will typically be applied for audio sent to internal speakers as well as audio sent to external S/PDIF (or other) audio outputs so that downstream amplifiers will also be in-sync.

A lipsync-aware Repeater will calculate the latency fields for its upstream EDID to indicate the overall video and audio latency from the reception by the Repeater to the eventual rendering by the Repeater or by downstream device(s). For instance, if the Repeater is a video processor the video data will be delayed by its internal processing before being passed downstream. In this case, the Repeater should indicate a video latency in the upstream EDID equal to the video latency found in the downstream device's EDID plus the Repeater's own internal video processing latency.

Likewise, if the Repeater is an audio amplifier which passes video through unmodified but which renders (amplifies) the audio directly (not passing it downstream), then the upstream audio latency will be equal to the Repeater's audio processing latency (only). If this amplifier adds an audio delay sufficient to compensate for the video latency of the downstream device, then the upstream audio and video latencies will be equal, whether that Repeater forwards the audio downstream or renders the audio directly.

8.9.1.1 Supporting a Range of Latency Values

If the video latency of a device differs significantly depending upon the video format or other factor, it is recommended that the video latency field indicate a latency that is between the extremes but skewed toward the longer latency. An audio/video mismatch is more perceptible if the audio leads the video than if the video leads the audio by a similar amount. Because of this effect, indicating a value that is closer to the maximum video delay may result in better overall user experience. For example, a value of roughly (2 * max_latency + min_latency)/3 may be used. The same is true for the audio latency but in this case, the indicated value should be skewed towards the minimum latency. For example, a value of roughly (max_latency + 2*min_latency)/3 may be used.

If the optimum indication for the video latency for interlaced video formats is significantly different than the optimum indication of latency for progressive formats, then the I_Latency_Fields_Present flag should be set, allowing the EDID to indicate separate latencies for these two categories of video formats. This approach may be used anytime but it is recommended in case the difference between the two latencies is more than roughly 30 msecs.

8.9.2 Compensation

A lipsync-aware Source or Repeater may delay the audio to compensate for the video latency of the downstream device(s), by an amount equal to the video_latency minus the audio_latency of the downstream (or rendering) devices.

It may not be possible to determine the audio latency of non-HDMI audio outputs (e.g. S/PDIF or analog outputs). For uncompressed audio formats, typically the value will be close to zero and so the device can simply delay the audio by the amount of video latency in the downstream EDID. For compressed audio formats, the device may assume that the audio latency is near the standard decompression latency specified in the relevant IEC61937-x standard or in the codec vendor's documentation.

It is expected that an audio delay capability of 100msecs will support full compensation for almost all of the TV and video processor products on the market today.

If transmitting a progressive video format, the Video_Latency and Audio_Latency fields are used for this calculation. If transmitting an interlaced video format, either these same fields are used or, if I_Latency_Fields_Present == 1, then the Interlaced_Video_Latency and Interlaced_Audio_Latency fields are used.

8.9.3 Supporting Dynamic Latency Changes

A future version of the HDMI Spec will define a mechanism for Sinks to dynamically modify their latency information.

8.9.4 Separate Audio and Video Paths

If the Source or Repeater splits the audio and video stream for transmission to two separate outputs, the device should calculate the required audio delay for the audio path by subtracting the audio latency in the audio path's EDID from the video latency in the video path's EDID.

A Source may use the no-video value or the corresponding no-audio value (in the Audio_Latency fields) to automatically determine whether video or audio is supported on a particular signal path.

The no-video value of the Video_Latency fields allows a Sink to declare that it does no internal rendering of the video signal nor does it output the HDMI-received video stream on an HDMI or non-HDMI output. This may never be true for a TV but it may be true or may change dynamically for an audio amplifier with a video pass-through function depending upon whether any device is connected downstream of the video output port, for example.

For a Repeater that does no video rendering, if there is no downstream video device connected to the Repeater's output, then the Repeater should indicate no-video. If there are downstream device(s) connected but all of those device(s) have no-video value in the video latency field then the Repeater should also indicate no-video.

9 Content Protection

9.1 Recommendation

Content protection capability is recommended for all HDMI compliant devices. An HDMI compliant Source should protect all of the protected audiovisual data. Amongst adequate copy protection technologies that are compatible with HDMI, HDCP is available.

9.2 HDCP Implementations

HDCP implementations for HDMI shall adhere to HDCP specification version 1.10.

Note that if the Sink has no digital audio outputs and has typical restrictions on its analog audio outputs (e.g. must be normal pitch) then it is recommended that Supports_AI be set. If this bit is clear then the Sink will not be able to receive audio content from DVD-Audio and Super Audio CD.

9.3 Usage of Audio Content Protection (ACP) Packets

A Source may use the ACP Packet to convey content-related information regarding the active audio stream.

Non-transmission of ACP Packets should be considered equivalent to transmission of an ACP Packet with an ACP_Type field of 0. If a Sink does not receive an ACP Packet within 600msecs, it shall revert to ACP_Type = 0 behavior.

Whenever a Source is required by other license agreements or specifications to transmit information related to the content protection requirements of the active audio stream, ACP Packets shall be transmitted at least once per 300msecs and an appropriate ACP_Type value shall be set.

When transmitting ACP Packets, upon the start of a new audio stream or upon any change in the audio stream that can be indicated by the ACP Packet, a modified, accurate ACP Packet shall be transmitted no later than 300msec following the transmission of the affected or relevant audio sample.

The ACP Packet transmission may occur at any time that a Data Island packet may be transmitted.

A Source shall not transmit an ACP Packet to a Sink that does not have Supports_AI = 1.

9.3.1 Requirements for Sink

A Sink that has any type of audio output and/or audio recording function shall be capable of receiving and appropriately handling the ACP Packet even if the Sink does not support any audio rendering functionality.

Whenever an HDCP-capable Sink detects an ACP Packet, it shall comply with the HDCP Audio Compliance Rules.

Whenever an HDCP-capable Sink detects an ACP Packet with an unknown ACP_Type value, it shall comply with the HDCP Audio Compliance Rules for undefined content.

9.3.2 Requirements for Repeater

Any content that is received by a Repeater and is accompanied by an ACP Packet shall be accompanied with an identical ACP Packet and any concurrently received ISRC1 or ISRC2 packets when that content is transmitted to a Sink with Supports_AI = 1.

9.3.3 Application to Generic Audio

With regards to the control of copying and audio output permissions, transmission of an ACP Packet with an ACP_Type field of 0 is equivalent to no transmission of an ACP Packet.

ACP_Type = 0 : Generic Audio

ACP_Type_Dependent fields all Reserved (0).

9.3.4 Application to IEC 60958-Identified Audio

A Source may indicate that the Sink must support the proper output of SCMS bits by setting ACP_Type = 1 (Type 1 = IEC 60958-identified).

ACP_Type = 1 : IEC 60958-identified

ACP_Type_Dependent fields all Reserved (0).

9.3.5 Application to DVD-Audio

Whenever a Source is transmitting DVD-Audio content for which HDCP is required, an accurate ACP Packet, with ACP_Type = 2 shall be transmitted at least once per 300msec.

The UPC/EAN and/or ISRC values are recorded on the DVD-Audio disc with DVD audio data. When the Source transmits UPC/EAN and/or ISRC using ISRC packet, the time lag between the ISRC packet and the corresponding audio sample packet should be minimized.

ACP_Type = 2 : DVD-Audio

ACP_Type_Dependent Usage:

Table 9-1 ACP_Type Dependent Fields for DVD-Audio Application

Packet Byte #	7	6	5	4	3	2	1	0
PB0	DVD-Audio_Type_Dependent_Generation							
PB1	Copy_Permission		Copy_Number			Quality		Transaction
PB2	Reserved (0)							
⋮								
PB27								

- DVD-Audio_Type_Dependent_Generation [8 bits] Identifies the generation of the DVD-Audio-specific ACP_Type_Dependent fields. Shall be set to 1. In the future version of this specification, currently reserved field(s) may be used to carry additional information. In such case, the value of this field may be incremented.
- Copy_Permission [2 bits] audio_copy_permission parameter.
- Copy_Number [3 bits] audio_copy_number parameter.
- Quality [2 bits] audio_quality parameter.
- Transaction [1 bit] audio_transaction parameter.

See “DVD Specifications for Read-Only Disc, Part 4: AUDIO SPECIFICATIONS”, Version 1.2, Table 7.2.3.1.1-2”, and “Supplement to Part 4: AUDIO SPECIFICATIONS Version 1.2 (February 2004)” for descriptions and use of the fields: audio_copy_permission, audio_copy_Number, audio_quality, and audio_transaction.

Any Source that supports DVD-Audio transmission on HDMI shall have the ability to transmit all valid channels of any multi-channel content.

9.3.6 Application to Super Audio CD

Whenever a Source is transmitting content originally derived from the HD Layer of a Super Audio CD, an accurate ACP Packet with ACP_Type = 3 shall be transmitted at least once per 300msec. See Super Audio CD System Description for “HD Layer Content”.

ACP_Type = 3 : Super Audio CD

ACP_Type_Dependent Usage:

Table 9-2 ACP_Type Dependent Fields for Super Audio CD Application

Packet Byte #	7	6	5	4	3	2	1	0
PB0	CCI_1							
: :								
PB15								
PB16	Reserved (0)							
: :								
PB27								

CCI_1 [16 bytes] Additional content control information. See Super Audio CD System Description for details.

Appendix A Repeater

A.1 Repeater Functions

A Repeater is defined as a device with one or more HDMI inputs, one or more HDMI outputs, and a retransmission function.

A Repeater has at least one of following functions:

- Repeat function:
  Single-input, single-output devices. Used primarily for cable extension.
- Switch function:
  Multiple-input, single-output devices. Used primarily to select among multiple Sources.
- Distributor function:
  Single-input, multiple-output devices, where only one output is active. Used primarily to select among multiple displays or Sinks.
- Duplicator function:
  Single-input, multiple-output devices, where more than one output is active. Used for signal distribution.

Combinations of the above, for instance, multiple-input, multiple-output devices, incorporating both input selection and output selection or signal distribution are allowed.

In all cases, each HDMI input shall fulfill all of the requirements of an HDMI Sink when it is connected with an active sink device, and each HDMI output shall fulfill all of the requirements of an HDMI Source when it is connected with an active source.

The E-EDID presented by a Repeater should reflect the capabilities of the downstream Sink.

A.2 E-EDID Read Timing (Informative)

In terms of E-EDID handling, Repeaters will typically fall into one of the following categories.

- Stored E-EDID type: The Repeater stores an E-EDID data structure that typically consists of downstream Sink capabilities.
- Forwarding type: The Repeater does not store an E-EDID data structure. When an E-EDID read request comes from a Source, the Repeater forwards the read request to a Sink. The E-EDID data from the Sink is then forwarded back to the Source.

An HDMI cluster may have several Repeaters between a Source and a Sink. To minimize the impact to the E-EDID reading time, each Repeater in the chain should minimize the added delay.

For example, the delay added by a Forwarding type Repeater should be no more than 4 msec per 16-byte read.

A stored E-EDID type Repeater should be able to send a 256 byte E-EDID within 150 msec when a Source issues sixteen 16-byte read requests. This means that a 16-byte read request would be completed within approximately 10 msec.

Appendix B Type B Connector Usage

B.1 Exception To Audio Format Support Requirement

Sources are not required to carry audio when all of the following conditions are met:

- Source is required by the HDMI Specification or associated agreements to use the Type B connector, and,
- Source has alternate default or user selectable audio outputs, and,
- Source can ensure that the appropriate audio stream is being delivered to the alternate audio outputs.

In order to guarantee rendering of video from Sources that do not fully support HDMI audio, the following condition shall be met:

- Sinks that are capable of supporting an HDMI video format when it is accompanied by audio shall also support that format when it is not accompanied by audio.

It is strongly recommended that a display device, when receiving an HDMI video signal without audio, temporarily indicate to the user that there is no audio accompanying the stream.

B.2 HDMI Dual-Link Architecture

HDMI dual-link architecture is compatible with DVI 1.0 dual-link architecture. Refer to section 3.1.5 of the DVI 1.0 specification.

Appendix C Compatibility With DVI

C.1 Requirement for DVI Compatibility

All HDMI Sources shall be compatible with DVI 1.0 compliant sink devices (i.e. “monitors” or “displays”) through the use of a passive cable converter. Likewise, all HDMI Sinks shall be compatible with DVI 1.0 compliant sources (i.e. “systems” or “hosts”) through the use of a similar cable converter.

When communicating with a DVI device, an HDMI device shall operate according to the DVI 1.0 specification, with the following exception – these devices are not required to comply with DVI 1.0 rules regarding:

- Monitor scaling requirements [refer to Section 2.2.8.2 of the DVI specification – superseded by HDMI specifications]
- Physical Interconnect specifications [refer to Chapter 5 of the DVI specification – superseded by HDMI specifications]
- System Low Pixel Format Support Requirements [refer to Section 2.2.4 of the DVI specification – superseded by HDMI specifications]

Furthermore, for HDMI devices that would not otherwise have a “BIOS” or “operating system” there are the following additional exceptions:

- “BIOS” requirements [refer to Section 2.2.4 of the DVI specification]
- “Operating system” requirements [refer to Section 2.2.2 and Section 2.2.9 of the DVI specification]
- “System level event” requirements [refer to Section 2.2.9.1 of the DVI specification]
- Power management requirements [refer to Section 2.4 of the DVI specification]

C.2 HDMI Source Requirements

When communicating with a DVI sink device, an HDMI Source shall operate in a mode compatible with that device. This requires that the Source operate under the following limitations:

- Video pixel encoding shall be RGB.
- No Video Guard Bands shall be used.
- No Data Islands shall be transmitted.

An HDMI Source may transmit Video Data Periods without Guard Bands only when communicating to a DVI sink device or during the process of determining if the sink device is HDMI capable.

An HDMI Source, upon power-up, reset or detection of a new sink device, shall assume that the sink device operates under DVI 1.0 limitations. An HDMI Source shall determine if the sink device is an HDMI Sink by following the rule(s) described in Section 8.3.3. Upon detection of an HDMI Sink, the HDMI Source shall follow all of the HDMI Source-related requirements specified in this document.

All electrical and physical specifications in Section 4 shall be followed by the HDMI Source even when communicating with a DVI sink device.

C.3 HDMI Sink Requirements

When connected to a DVI source device, an HDMI Sink shall operate as a DVI 1.0 compliant sink with the exceptions outlined in Section C.1 above.

A DVI source device will always be restricted in the following ways:

- Only RGB pixel encoding is used.
- There is no Guard Band on the Video Data Period.
- There are no Data Islands transmitted.

An HDMI Sink, upon power-up, reset or detection of a new source device, shall assume that the source device is limited to the above behavior. Upon the detection of an indication that the source is HDMI-capable, the HDMI Sink shall follow all of the HDMI Sink-related requirements specified in this document.

All electrical and physical specifications in Section 4 of the HDMI Specification shall be followed by the HDMI Sink even when communicating with a DVI source device.

C.4 Type A to DVI Adapter Cable [Informative]

Table C-1 Wire Categories

Category	Description
A	TMDS Signal Wire
B	TMDS Shield
C	Control
D	Control Ground
N.C.	No connect (no wire)
5V	5 Volts Power Wire

Table C-2 Type A-to-DVI-D Cable Wire Assignment [Informative]

Type A pin	Signal Name	Wire	DVI-D pin
1	TMDS Data2+	A	2
2	TMDS Data2 Shield	B	3
3	TMDS Data2–	A	1
4	TMDS Data1+	A	10
5	TMDS Data1 Shield	B	11
6	TMDS Data1–	A	9
7	TMDS Data0+	A	18
8	TMDS Data0 Shield	B	19
9	TMDS Data0–	A	17
10	TMDS Clock+	A	23
11	TMDS Clock Shield	B	22
12	TMDS Clock–	A	24
15	SCL	C	6
16	DDC Data	C	7
17	DDC/CEC Ground	D	15
18	+5V Power	5V	14
19	Hot Plug Detect	C	16
13	CEC	N.C.	
14	Reserved (in cable but N.C. on device)	N.C.	
	TMDS Data 4-	N.C.	4
	TMDS Data 4+	N.C.	5
	TMDS Data 3-	N.C.	12
	TMDS Data 3+	N.C.	13
	TMDS Data 5-	N.C.	20
	TMDS Data 5+	N.C.	21
	No Connect	N.C.	8

C.5 Type B to DVI Adapter Cable [Informative]

Table C-3 Type B to DVI-D Cable Wire Assignment [Informative]

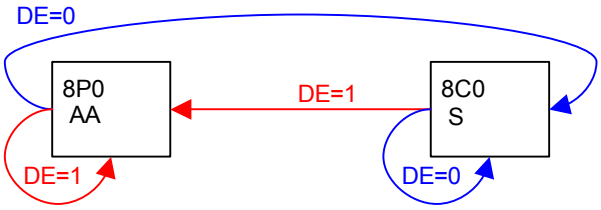
Type B pin	Pin Assignment	Wire	DVI-D pin
1	TMDS Data2+	A	2
2	TMDS Data2 Shield	B	3
3	TMDS Data2-	A	1
4	TMDS Data1+	A	10
5	TMDS Data1 Shield	B	11
6	TMDS Data1-	A	9
7	TMDS Data0+	A	18
8	TMDS Data0 Shield	B	19
9	TMDS Data0-	A	17
10	TMDS Clock+	A	23
11	TMDS Clock Shield	B	22
12	TMDS Clock-	A	24
13	TMDS Data5+	A	21
14	TMDS Data5 Shield	B	19
15	TMDS Data5-	A	20
16	TMDS Data4+	A	5
17	TMDS Data4 Shield	B	3
18	TMDS Data4-	A	4
19	TMDS Data3+	A	13
20	TMDS Data3 Shield	B	11
21	TMDS Data3-	A	12
25	SCL	C	6
26	DDC Data	C	7
27	DDC/CEC Ground	D	15
28	+5V Power	5V	14
29	Hot Plug Detect	C	16
22	CEC	N.C.	
23	Reserved	N.C.	
24	Reserved	N.C.	
	No Connect	N.C.	8

Appendix D Deep Color Additional Detail

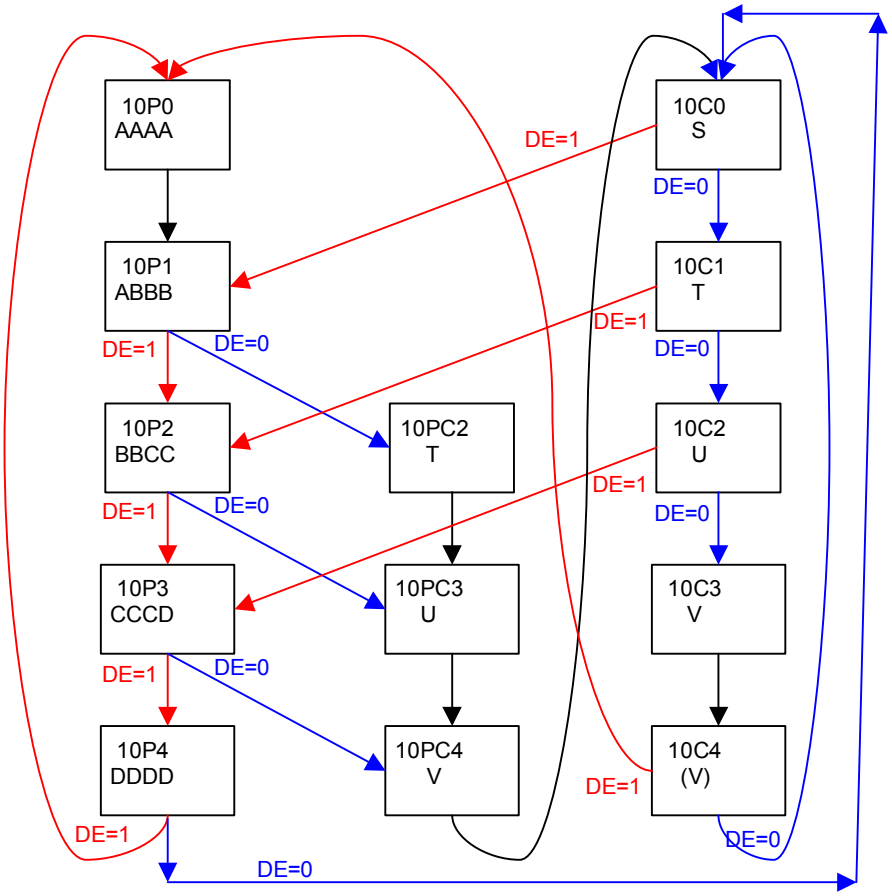
E.1 State Machines

The following state machine drawings are provided for informative purposes to provide better understanding of the Deep Color packing sequences. For each mode, the source sequence starts at phase 0, and then increments through the phases. While DE=1 (pixel data is available), pixel data fragments *m*P*n* are transmitted. While DE=0, blanking fragments *m*C*n* are transmitted.

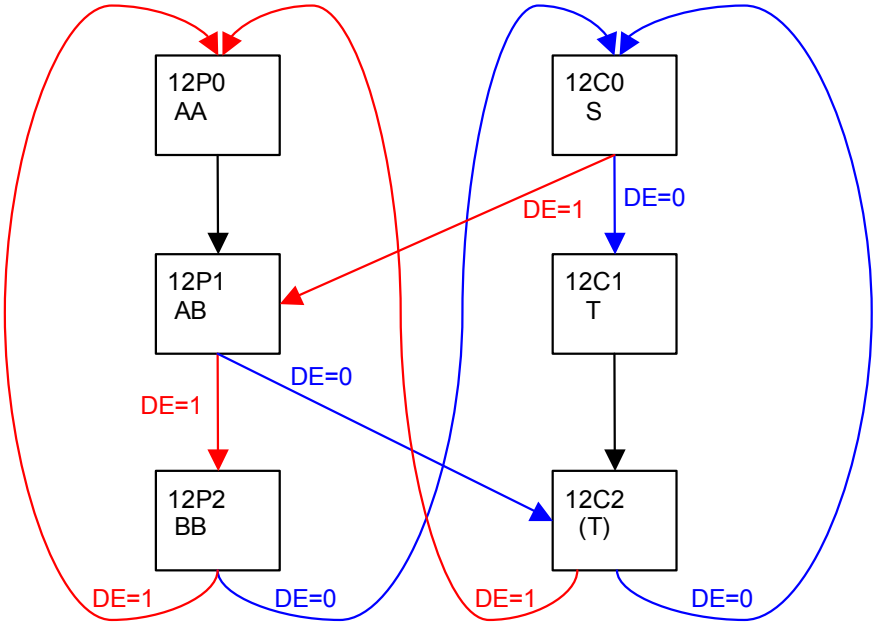
24 bit mode:



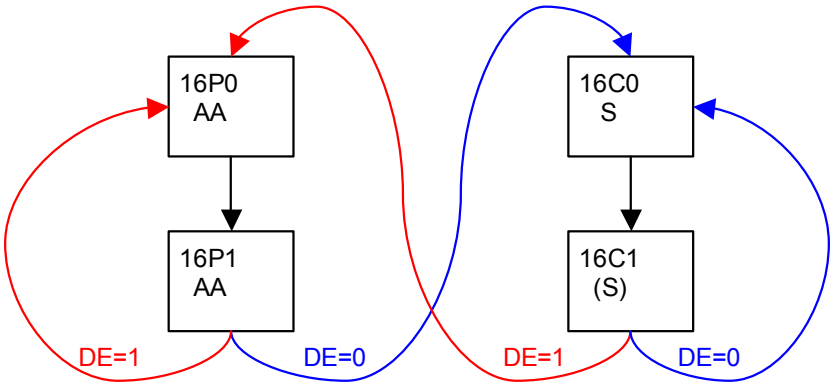
30 bit mode:



36 bit mode:



48 bit mode:



D.2 Recommended N and Expected CTS Values

The recommended value of N for several standard pixel clock rates at several Deep Color modes are shown below. It is recommended that Sources with non-coherent clocks use the values listed for a TMDS clock of “Other”.

Table D-1 36 bits/pixel: Recommended N and Expected CTS for 32kHz

	32 kHz	
TMDS Clock (MHz)	N	CTS
25.2 / 1.001	9152	84375
25.2	4096	37800
27	4096	40500
27 * 1.001	8192	81081
54	4096	81000
54 * 1.001	4096	81081
74.25 / 1.001	11648	316406-316407*
74.25	4096	111375
148.5 / 1.001	11648	632812-632813*
148.5	4096	222750
Other	4096	Measured

** Note: This value will alternate because of restriction on N.*

Table D-2 36 bits/pixel: Recommended N and Expected CTS for 44.1kHz and Multiples

	44.1 kHz		88.2 kHz		176.4 kHz	
TMDS Clock (MHz)	N	CTS	N	CTS	N	CTS
25.2 / 1.001	7007	46875	14014	46875	28028	46875
25.2	6272	42000	12544	42000	25088	42000
27	6272	45000	12544	45000	25088	45000
27 * 1.001	6272	45045	12544	45045	25088	45045
54	6272	90000	12544	90000	25088	90000
54 * 1.001	6272	90090	12544	90090	25088	90090
74.25 / 1.001	17836	351562-351563*	35672	351562-351563*	71344	351562-351563*
74.25	6272	123750	12544	123750	25088	123750
148.5 / 1.001	17836	703125	35672	703125	71344	703125
148.5	6272	247500	12544	247500	25088	247500
Other	6272	measured	12544	measured	25088	measured

** Note: This value will alternate because of restriction on N.*

Table D-3 36 bits/pixel: Recommended N and Expected CTS for 48kHz and Multiples

	48 kHz		96 kHz		192 kHz	
TMDS Clock (MHz)	N	CTS	N	CTS	N	CTS
25.2 / 1.001	9152	56250	18304	56250	36608	56250
25.2	6144	37800	12288	37800	24576	37800
27	6144	40500	12288	40500	24576	40500
27 * 1.001	8192	54054	16384	54054	32768	54054
54	6144	81000	12288	81000	24576	81000
54 * 1.001	6144	81081	12288	81081	24576	81081
74.25 / 1.001	11648	210937-210938*	23296	210937-210938*	46592	210937-210938*
74.25	6144	111375	12288	111375	24576	111375
148.5 / 1.001	11648	421875	23296	421875	46592	421875
148.5	6144	222750	12288	222750	24576	222750
Other	6144	measured	12288	measured	24576	measured

** Note: This value will alternate because of restriction on N.*

Table D-4 48 bits/pixel: Recommended N and Expected CTS for 32kHz

	32 kHz	
TMDS Clock (MHz)	N	CTS
25.2 / 1.001	4576	56250
25.2	4096	50400
27	4096	54000
27 * 1.001	4096	54054
54	4096	108000
54 * 1.001	4096	108108
74.25 / 1.001	11648	421875
74.25	4096	148500
148.5 / 1.001	11648	843750
148.5	4096	297000
Other	4096	Measured

Table D-5 48 bits/pixel: Recommended N and Expected CTS for 44.1kHz and Multiples

	44.1 kHz		88.2 kHz		176.4 kHz	
TMDS Clock (MHz)	N	CTS	N	CTS	N	CTS
25.2 / 1.001	7007	62500	14014	62500	28028	62500
25.2	6272	56000	12544	56000	25088	56000
27	6272	60000	12544	60000	25088	60000
27 * 1.001	6272	60060	12544	60060	25088	60060
54	6272	120000	12544	120000	25088	120000
54 * 1.001	6272	120120	12544	120120	25088	120120
74.25 / 1.001	17836	468750	35672	468750	71344	468750
74.25	6272	165000	12544	165000	25088	165000
148.5 / 1.001	8918	468750	17836	468750	35672	468750
148.5	6272	330000	12544	330000	25088	330000
Other	6272	measured	12544	measured	25088	measured

Table D-6 48 bits/pixel: Recommended N and Expected CTS for 48kHz and Multiples

TMDS Clock (MHz)	48 kHz		96 kHz		192 kHz	
	N	CTS	N	CTS	N	CTS
25.2 / 1.001	6864	56250	13728	56250	27456	56250
25.2	6144	50400	12288	50400	24576	50400
27	6144	54000	12288	54000	24576	54000
27 * 1.001	6144	54054	12288	54054	24576	54054
54	6144	108000	12288	108000	24576	108000
54 * 1.001	6144	108108	12288	108108	24576	108108
74.25 / 1.001	11648	281250	23296	281250	46592	281250
74.25	6144	148500	12288	148500	24576	148500
148.5 / 1.001	5824	281250	11648	281250	23296	281250
148.5	6144	297000	12288	297000	24576	297000
Other	6144	measured	12288	measured	24576	measured

Appendix E Gamut-Related Metadata

E.1 Overview

Unlike the classic colorimetry standards used for standard definition and high definition video, the enhanced colorimetry of xvYCC has a virtually unconstrained gamut, which does not easily map to the real-world gamut of existing or near-future display technologies. When transmitting such an enhanced colorimetry, it is necessary for the video source to also indicate the color gamut of the transmitted video. This metadata allows the display to map the gamut of the video stream more accurately and more predictably onto the gamut of the display.

The gamut describes all colors that are reproducible by a particular reference display or that are present in a given content. The color gamut can be described by a Gamut Boundary Description (GBD). When a given image content has a gamut larger or different from the gamut of the HDMI sink, the colors lying outside the aimed gamut need to be clipped or moved accordingly. This procedure is called gamut mapping. The gamut of the content is circumscribed by the gamut of the HDMI source color space. The display can then use the content's Gamut Boundary Description to perform accurate and predictable mapping onto its own gamut.

E.2 Transmission Profiles

There are several transmission profiles (P0, P1, etc.) for gamut related metadata. The difference between the transmission profiles is primarily the transmission rate, specifically, the number of packets that may be sent per video field. Because of the need to transmit the entire metadata within a short period of time, this transmission rate limits the maximum size of the profile as well. The maximum size of the metadata then also corresponds to the accuracy of the gamut boundary description.

The lowest speed transmission profile is P0, transmitting at a rate of one Gamut Metadata Packet per video field. P0 metadata fits completely within that one packet.

When transmitting GBD data, the Source shall send the GBD for an upcoming gamut change using P0 transmission profile if that GBD is available prior to transmission of the content. This P0 transmission should occur at least one full video field prior to the start of the new gamut video.

E.3 Gamut Boundary Description

The HDMI source gamut is described either by a set of R/G/B range limits or by a set of vertices with or without indexed facets. The Format_Flag field indicates which format is supplied:

- Format_Flag [1 bit] Identifies whether subsequent data describes gamut range boundary or gamut vertices boundary. A value of 0 indicates vertices/facets description. 1 indicates range description.

Two simple GBD data structures: one based on four vertices and the other based on R/G/B min/max range limits, are fully defined here and may be supported using the P0 transmission profile. Larger data structures carrying many more vertices as well as facets, will be defined in a future international specification, not necessarily the HDMI Specification. These data structures will require higher transmission speeds and sizes and therefore will require higher transmission profiles (P1...).

After those data structures are defined, a subsequent version of the HDMI Specification will permit HDMI devices to support them. Until that time, no HDMI Source may transmit a GBD with

more than four vertices or containing facet data, nor may any Sink indicate support for transmission profiles other than P0 in the Colorimetry Data Block in the EDID.

Facet data descriptions below are informative only. The facet data structures and relationship between facets and vertices will be described in a future specification.

The size of each component of each vertex or range entry is indicated in the field GBD_Color_Precision:

- GBD_Color_Precision [2 bits] Color precision of GBD vertex and range data:
  0b00 8 bit
  0b01 10 bit
  0b10 12 bit

The definition of vertex data depends on the GBD_Color_Space field (see below) and is as follows:

- RGB: Unsigned integer. According to ITU-R 709-5 item 6.10 (8 bit), summarized in Section 6.6 (for 8**...**16 bit).
- xvYCC: Unsigned integer. According to IEC 61966-2-4 item 4.4, summarized in Section 6.6
- XYZ: Not valid. Future versions of this specification will further define XYZ tristimulus. Until that point, XYZ shall not be used for GBD data.

The precision of the facet data (Packed_GBD_Facets_Data in Table E-1) depends on the number of vertices (Number_Vertices) according to the following equation:

Precision [number of bits] of color facet data = ld(Number_Vertices)

Where ld() is the logarithm to the base of two. The format of color facet data is positive integer, each color facet data indicating the index of a vertex in the Packed_GBD_Vertices_data field. For example, the integer 0 indicates the first vertex in Packed_GBD_Vertices_data and the integer 1 indicates the second vertex in Packed_GBD_Vertices_data. Three consecutive facets data define one triangle with surface normal pointing outside the gamut.

The definition of 8/10/12-bit range data is as follows:

- 8-bit: signed fixed-point – 1 sign bit, 2 bits integer, 5 bits fraction
- 10-bit: signed fixed-point – 1 sign bit, 2 bits integer, 7 bits fraction
- 12-bit: signed fixed-point – 1 sign bit, 2 bits integer, 9 bits fraction

The data structures for these two different formats are shown in Table E-1 and Table E-2 .

Table E-1 Vertices/Facets GBD Data Structure

Byte #	7	6	5	4	3	2	1	0
0	Format_ Flag = 0	Facet_ Mode	Rsvd(0)	GBD_Color_Precision		GBD_Color_Space		
1	Number_Vertices_H							
2	Number_Vertices_L							
3...VSIZE+2	Packed_GBD_Vertices_Data[0... VSIZE-1]							
VSIZE+3	Number_Facets_H							
VSIZE+4	Number_Facets_L							
VSIZE+5...	Packed_GBD_Facets_Data[0...FSIZE-2]							
VSIZE+FSIZE+4	Packed_GBD_Facets_Data[FSIZE-1]							

- Facet_Mode [1 bit] Indicates if Facets are also included in the GBD. Field is valid only when Format_Flag = 0. Reserved (0) when Format_Flag = 1.
- GBD_Color_Precision [2 bits] see above
- GBD_Color_Space [3 bits] Color space of GBD data:
  0b000 ITU-R BT.709 (using RGB)
  0b001 xvYCC$_{601}$ (IEC 61966-2-4 – SD) (using YC_BC_R)
  0b010 xvYCC$_{709}$ (IEC 61966-2-4 – HD) (using YC_BC_R)
  0b011 XYZ (see above)
- Number_Vertices(_H, _L) [2 bytes] Number of vertices described by following structure.
- Number_Facets(_H, _L) [2 bytes] Number of facets described by following structure.

VSIZE is the number of bytes in the Packed_GBD_Vertices_Data according to:

$$VSIZE = INT\left(3 \cdot Number\_Vertices \cdot GBD\_Color\_Precision / 8 + 0.99999\right)$$

Where, INT() is a function returning the integer part of the number (e.g. INT(3.99999...) = 3).

FSIZE is the number of bytes of Packed_GBD_Facets_Data and will be defined in a future specification.

The minimal number of vertices is 4. In this case, and only this case, the vertices have the following meaning, in this order: black point, red primary, green primary and blue primary. This convention allows constructing the white point and the secondary colors (magenta, cyan and yellow) without transmission.

Table E-2 Range GBD Data Structure

Byte #	7	6	5	4	3	2	1	0
0	Format_ Flag = 1	Rsvd(0)	Rsvd(0)	GBD_Color_Precision		GBD_Color_Space		
1...N	Packed_Range_Data							

- Packed_Range_Data [N bytes] Packed range data according to following sequence:
  Min_Red_Data
  Max_Red_Data
  Min_Green_Data
  Max_Green_Data
  Min_Blue_Data
  Max_Blue_Data
- GBD_Color_Precision [2 bits] see above
- GBD_Color_Space [3 bits] Color space of GBD data:
  0b000 Reserved
  0b001 RGB expression of $xvYCC_{601}$ coordinates
  0b010 RGB expression of $xvYCC_{709}$ coordinates
  0b011 Reserved

E.4 Data Packing

GBD data is efficiently packed with each 8-, 10- and 12-bit value taking exactly 8-, 10- or 12-bits in the packet. The GBD_Color_Precision field specifies the packing and precision of the GBD data. Table E-3 and Table E-4 define the packing for 10- and 12-bit values using a representative sequence of values, A, B, C..., with A_low representing the low-order bits and A_high, the high-order bits of value A.

Table E-3 10-bit Packing

	7	6	5	4	3	2	1	0
0	A_high							
1	A_low			B_high				
2	B_low				C_high			
3	C_low					D_high		
4	D_low							
5	E_high							
6	E_low			F_high				
7	F_low				G_high...			

Table E-4 12-bit Packing

	7	6	5	4	3	2	1	0
0	A_high							
1	A_low				B_high			
2	B_low							
3	C_high							
4	C_low				D_high			
5	D_low							

E.5 Example P0 Data Structures

A simple but useful vertex GBD data structure is defined in Table E-5 and can be transmitted using a single Gamut Metadata Packet, fitting within the P0 transmission profile.

The gamut is described in xvYCC$_{709}$ space at 8-bit. The GBD consists of black point as well as red, green and blue primaries.

This data structure has the minimum number of vertices. Following the specification, the correspondence of transmitted vertices and primaries and black point is given. To reconstruct the full gamut boundary description, the white point vertex V_{WHITE} and the secondary colors ($V_{MAGENTA}, V_{CYAN}, V_{YELLOW}$ for magenta, cyan and yellow, respectively) are generated from the first primaries as follows:

$$V_{WHITE} = V_{RED} + V_{GREEN} + V_{BLUE} - 2V_{BLACK}$$
$$V_{MAGENTA} = V_{RED} + V_{BLUE} - V_{BLACK}$$
$$V_{CYAN} = V_{GREEN} + V_{BLUE} - V_{BLACK}$$
$$V_{YELLOW} = V_{RED} + V_{GREEN} - V_{BLACK}$$

Table E-5 P0 Vertices-Only Data – 8-bit Precision Example

Byte #	7	6	5	4	3	2	1	0
0	Format_ Flag = 0	Facet_ Mode = 0	Rsvd = 0	GBD_Color_Precision = 00		GBD_Color_Space = 010		
1	Number_Vertices_H = 0							
2	Number_Vertices_L = 4							
3	Black(Y)							
4	Black(C_B)							
5	Black(C_R)							
6	Red(Y)							
7	Red(C_B)							
8	Red(C_R)							
9	Green(Y)							
10	Green(C_B)							
11	Green(C_R)							
12	Blue(Y)							
13	Blue(C_B)							
14	Blue(C_R)							

A typical range GBD data structure is shown in Table E-6 . This can be transmitted using a single Gamut Metadata Packet, fitting within the P0 transmission profile. The gamut is described in $xvYCC_{709}$ space at 12-bit. The GBD consists of min_red_data, max_red_data, min_green_data, max_green_data, min_blue_data, and max_blue_data

Table E-6 P0 Range Data – 12-bit Precision Example

Byte #	7	6	5	4	3	2	1	0
0	Format_ Flag = 1	Rsvd = 0	Rsvd = 0	GBD_Color_Precision = 10		GBD_Color_Space = 010		
1	Min_Red_Data_H							
2	Min_Red_Data_L				Max_Red_Data_H			
3	Max_Red_Data_L							
4	Min_Green_Data_H							
5	Min_Green_Data_L				Max_Green_Data_H			
6	Max_Green_Data_L							
7	Min_Blue_Data_H							
8	Min_Blue_Data_L				Max_Blue_Data_H			
9	Max_Blue_Data_L							

Appendix F Video Scaling Auto-Configuration

Informative Appendix

In addition to any other mode that a source provides, it is highly recommended that it also provide a pass-through mode of operation. In the pass-through mode the source will pass video unprocessed to its output, if the format is supported by the display, without scaling or de-interlacing (except for performing any necessary field repeats, for instance to show 720p24Hz content on a display with only a 720p60Hz input capability).

In the case that the source cannot send the video in pass-through mode (because the format is not supported by the display), it should convert to the highest priority format as indicated by the DTDs and SVDs, with the first DTD (“Preferred”) being the highest priority.

In the case that the source cannot send the video in pass-through mode (format not supported by the display), and it can also not convert to the preferred video format, the source should select the highest resolution progressive video format supported by the display.

Note: in order to allow displays to indicate a wide range of supported video formats, the source must be able to read EDID information from all defined blocks and must read and understand DTDs and SVDs as defined in CEA-861-D.

The source may also provide a “film-mode” de-interlacer to convert interlaced format video to its original progressive format. It should then consider such converted video as progressive, for instance 480i 60Hz video should be considered as 480p 24Hz video after successful film mode conversion.

It is strongly recommended that displays that cannot perform film-mode de-interlacing on an interlaced video format do not list such an interlaced format as the preferred format but list such a format with a priority (in the list of DTDs and SVDs) that corresponds to the effective resolution. For instance, a display that cannot do film-mode de-interlacing on 1080i may list this format with a priority roughly equivalent to 540 progressive lines.

Supplement 1

Consumer Electronics Control (CEC)

DELL/HP/LENOVO EXHIBIT 1015
Page 526

CEC Table of Contents

CEC Page

CEC 1 INTRODUCTION 1

CEC 1.1 Normative references 1

CEC 1.2 Informative References 1

CEC 1.3 Document Revision History 1

CEC 2 DEFINITIONS 2

CEC 2.1 Conformance Levels 2

CEC 2.2 Glossary of Terms 2

CEC 2.3 Usages and Conventions 2
CEC 2.3.1 State Diagrams 2
CEC 2.3.2 Message Flow Diagrams 3
CEC 2.3.3 Notation 3

CEC 3 FEATURE OVERVIEW 4

CEC 3.1 End-User Features 4

CEC 3.2 Supporting Features 4

CEC 4 ELECTRICAL SPECIFICATION 5

CEC 5 SIGNALING AND BIT TIMINGS 7

CEC 5.1 CEC Line Usage 7

CEC 5.2 Bit Timing 7
CEC 5.2.1 Start Bit Timing 7
CEC 5.2.2 Data Bit Timing 7

CEC 6 FRAME DESCRIPTION 10

CEC 6.1 Header/Data Block description 10
CEC 6.1.1 EOM (End of Message) 10
CEC 6.1.2 ACK (Acknowledge) 10
CEC 6.1.3 Header Block Details 11

CEC 7 RELIABLE COMMUNICATION MECHANISMS 12

CEC 7.1 Frame Re-transmissions 12

CEC 7.2 Flow Control 12

CEC 7.3 Frame Validation 12

CEC 7.4 CEC Line Error Handling 12

CEC 8 PROTOCOL EXTENSIONS 13

CEC 9 CEC ARBITRATION 14

DELL/HP/LENOVO EXHIBIT 1015
Page 527

CEC 9.1 Signal Free Time 14

CEC 9.2 Message Time Constraints 14

CEC 10 DEVICE CONNECTIVITY AND ADDRESSING 15

CEC 10.1 Physical Address Discovery 15

CEC 10.2 Logical Addressing 15
CEC 10.2.1 Logical Address Allocation 16

CEC 11 SWITCH REQUIREMENTS 18

CEC 11.1 CEC Switch 18

CEC 11.2 Non-CEC Switch 18
CEC 11.2.1 Switches with One EDID 18
CEC 11.2.2 Switches with Multiple EDIDs 19

CEC 12 HIGH LEVEL PROTOCOL 20

CEC 12.1 Source Declaration 20

CEC 12.2 Protocol General Rules 20

CEC 12.3 Feature Abort 20

CEC 12.4 Abort 21

CEC 13 CEC FEATURES DESCRIPTION 22

CEC 13.1 One Touch Play 22
CEC 13.1.1 Messages 22
CEC 13.1.2 Feature Description 22

CEC 13.2 Routing Control 23
CEC 13.2.1 Messages 23
CEC 13.2.2 Feature Description 23

CEC 13.3 System Standby 24
CEC 13.3.1 Messages 24
CEC 13.3.2 Feature Description 24

CEC 13.4 One Touch Record 26
CEC 13.4.1 Messages 26
CEC 13.4.2 Feature Description 26

CEC 13.5 System Information 27
CEC 13.5.1 Messages 27
CEC 13.5.2 Feature Description 27
CEC 13.5.3 Additional Information 28

CEC 13.6 Deck Control 28
CEC 13.6.1 Messages 28
CEC 13.6.2 Feature Description 28

CEC 13.7 Tuner Control 29
CEC 13.7.1 Messages 29

DELL/HP/LENOVO EXHIBIT 1015
Page 528

CEC 13.7.2 Feature Description....29

CEC 13.8 Vendor Specific Commands....30
CEC 13.8.1 Messages....30
CEC 13.8.2 Feature Description....30

CEC 13.9 OSD Status Display....31
CEC 13.9.1 Messages....31
CEC 13.9.2 Feature Description....31

CEC 13.10 Device OSD Name Transfer....31
CEC 13.10.1 Messages....31
CEC 13.10.2 Feature Description....32

CEC 13.11 Device Menu Control....32
CEC 13.11.1 Messages....32
CEC 13.11.2 Feature Description....32

CEC 13.12 Remote Control Pass Through....32
CEC 13.12.1 Messages....32
CEC 13.12.2 Feature Description....33

CEC 13.13 Give Device Power Status....33
CEC 13.13.1 Messages....33
CEC 13.13.2 Feature Description....33

CEC 14 DEVICE STATES....34
CEC 14.1.1 Device States....34
CEC 14.1.2 State Changes....34
CEC 14.1.3 All Devices....34
CEC 14.1.4 TV....34
CEC 14.1.5 Recording Device....35
CEC 14.1.6 Playback Device....36
CEC 14.1.7 Menu Providing Device....36

CEC 15 MESSAGE DESCRIPTIONS....37

CEC 16 MESSAGE DEPENDENCIES....50

CEC 17 OPERAND DESCRIPTIONS....55

DELL/HP/LENOVO EXHIBIT 1015
Page 529

CEC Figures

CEC Figure 1 Example State Diagram 3
CEC Figure 2 Example Message Flow Diagram 3
CEC Figure 3 Start bit pulse format showing minimum and maximum tolerances 7
CEC Figure 4 Timing diagrams for both bit states 8
CEC Figure 5 Timing Diagram for Follower Asserted Bit (Logical 0) 9
CEC Figure 6 Block Structure 10
CEC Figure 7 Header Block 11
CEC Figure 8 Logical Address Allocation 17
CEC Figure 9 Behavior when beneath a one EDID non-CEC Switch 19
CEC Figure 10 A typical scenario illustrating the One Touch Play feature 22
CEC Figure 11 Example message flow, when a CEC Switch is manually switched 24
CEC Figure 12 A typical scenario for the Standby feature to a specific device 25
CEC Figure 13 A typical scenario for the broadcast (system) Standby feature 25
CEC Figure 14 A typical scenario for the One Touch Record feature 26
CEC Figure 15 Message exchange when getting the TV's menu Language 27
CEC Figure 16 A typical scenario when a menu language setting within the TV is modified 28
CEC Figure 17 A typical scenario for the Deck Control feature 29
CEC Figure 18 A typical scenario for selecting a new Digital Service 30
CEC Figure 19 The messages sent in the Vendor Specific Commands feature 31

DELL/HP/LENOVO EXHIBIT 1015
Page 530

CEC Tables

CEC Table 1 CEC Electrical Specifications during the fully powered-Off state 5
CEC Table 2 CEC Electrical Specifications except during the fully powered-Off state 6
CEC Table 3 Frame Description 10
CEC Table 4 Signal Free Time 14
CEC Table 5 Logical Addresses 16
CEC Table 6 Message Descriptions for the One Touch Play Feature 38
CEC Table 7 Message Descriptions for the Routing Control Feature 39
CEC Table 8 Message Descriptions for the Standby Feature 40
CEC Table 9 Message Descriptions for the One Touch Record Feature 41
CEC Table 10 Message Descriptions for the System Information Feature 42
CEC Table 11 Message Descriptions for the Deck Control Feature 43
CEC Table 12 Message Descriptions for the Tuner Control Feature 44
CEC Table 13 Message Descriptions for the Vendor Specific Commands Feature 45
CEC Table 14 Message Descriptions for the OSD Status Display Feature 46
CEC Table 15 Message Descriptions for the Device OSD Transfer Feature 46
CEC Table 16 Message Descriptions for the Device Menu Control Feature 47
CEC Table 17 Message Descriptions for the Remote Control Passthrough Feature 48
CEC Table 18 Message Descriptions for the Power Status Feature 48
CEC Table 19 Message Descriptions for General Protocol messages 49
CEC Table 20 Message dependencies when receiving a message 50
CEC Table 21 Message dependencies when sending a message 53
CEC Table 22 Operand Descriptions. 55
CEC Table 23 User Control Codes 62

CEC 1 Introduction

CEC is a protocol that provides high-level control functions between all of the various audiovisual products in a user's environment. This appendix describes the CEC protocol in the following order:

- An overview of the recommended features available in CEC.
- A Low Level Protocol Definition - Includes the electrical specification, signaling and bit timings and the frame description.
- A High Level Protocol Definition - Includes a detailed feature breakdown and individual message descriptions.

CEC 1.1 Normative references

[1n] ISO 639.2 Code for the representation of names of languages - Part 2: Alpha 3 code
http://www.loc.gov/standards/iso639-2/langhome.html

CEC 1.2 Informative References

[1i] CENELEC, EN 50049-1:1997/A1:1998, Domestic and similar electronic equipment interconnection requirements: Peritelevision connector

[2i] CENELEC, EN 50157, Domestic and similar electronic equipment interconnection requirements: AV.link
EN 50157-1 : Part 1
EN 50157-2-1 : Part 2-1
EN 50157-2-2 : Part 2-2
EN 50157-2-3 : Part 2-3

[3i] IEEE std. 1394-1995 HIGH PERFORMANCE SERIAL BUS section 8.3.2.5.1 – example use of Company_id.

CEC 1.3 Document Revision History

1.2 Clarification of CEC line Standby behaviour
Clarification of test conditions in Table 2
Addition of CEC line pull-up using a current source
Addition of Give Power Status message
Clarification of response to <Abort> message

1.2A Tolerance on internal pull-up resistance changed to ±5% in Table 2.
Removal of test conditions from Table 2,
Clarification of maximum message length.
Re-ordering of some Features in the text and splitting of message description table.
Update and clarification of mandatory and optional implementation status.
Clarification of rules with more explanations for Routing Control.
Additional examples and notes regarding the use of System Standby with recordings.
System Info simplified to language selection. <Set Language> now becomes <Set Menu Language> with a simplified mechanism. Removal of <Set System Info Version Number>, <Give System Info> and <Set Country>.
Removal of analogue tuning messages and addition of <Set Digital Service>.
Removal of Preset Download and Timer Programme Features.
Various editorial corrections throughout

1.3 No Changes.

DELL/HP/LENOVO EXHIBIT 1015
Page 532

CEC 2 Definitions

CEC 2.1 Conformance Levels

Because CEC is optional, the conformance level in this appendix is only effective when the device supports CEC. For example, the word "shall" indicates a mandatory requirement for the CEC supporting devices. However, within the Features section (CEC 13) "shall" only indicates a requirement if the feature is implemented.

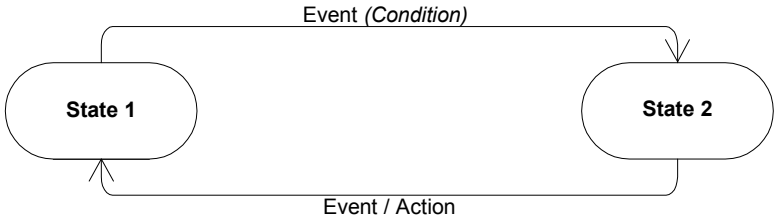
CEC 2.2 Glossary of Terms

Broadcast Message	This is a message, sent to logical address 15, on which all devices are expected to receive.
Clear	Set to an empty/undefined state. When a physical address is cleared it takes the value F.F.F.F. When a logical address is cleared it takes the value 15.
Deck	The part of a recording device or playback device that provides playback functionality usually from a removable media.
Destination	The target device for a CEC message.
Follower	A device that has just received a CEC message and is required to respond to it.
Initiator	The device that is sending, or has just sent, a CEC message and, if appropriate, is waiting for a follower to respond.
Logical address	A unique address assigned to each device (see section CEC 10.2)
Menu Providing Device	A non-display device that may render a menu on TV.
Playback device	A device that has the ability to play media, e.g. a DVD Player.
Recording device	A device that has the ability to record a source such as an internal digital tuner or an external connection.
Source Device	A device that is currently providing an AV stream via HDMI.
Tuner Device	A device that contains a digital tuner, e.g. a STB or a Recording Device.
Timer Setting Device	A device that has the ability to set the record timer blocks of a recording device.
TV	A device with HDMI input that has the ability to display the input HDMI signal. Generally it has no HDMI output.

CEC 2.3 Usages and Conventions

CEC 2.3.1 State Diagrams

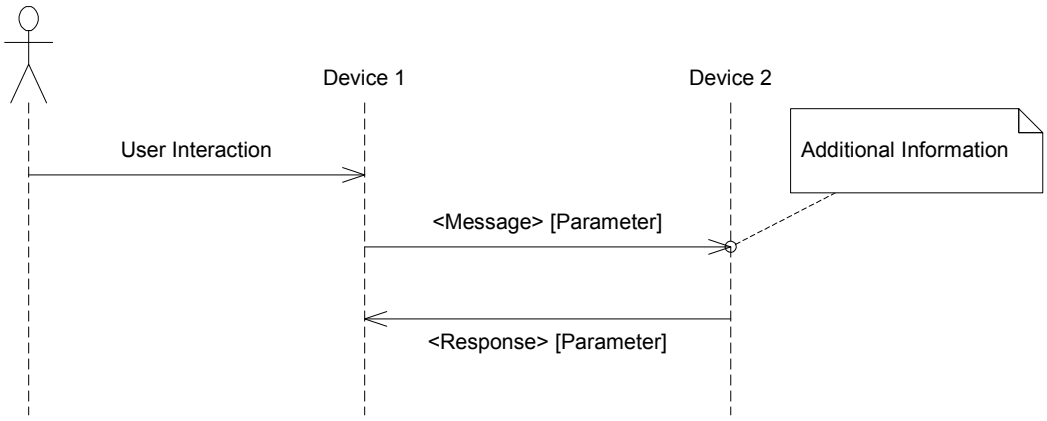
State diagrams describe behavior in terms of device states and events or actions. In these diagrams, the ovals represent device states and the arrows represent events and/or actions that move the device from one state to another state.



CEC Figure 1 Example State Diagram

CEC 2.3.2 Message Flow Diagrams

Message Flow Diagrams show sequences of messages that occur between 2 devices.



CEC Figure 2 Example Message Flow Diagram

CEC 2.3.3 Notation

Within the CEC specification there are a number of notations:

<xxx> xxx is an opcode for a message, which is defined in section CEC 15

[yyy] yyy is a data item, which is defined in section CEC 17.

“zzz” zzz is a constant and is a possible value for a data item in section CEC 17.

N{….} indicates the item within the braces is repeated N times, this is used mainly in section CEC 17.

DELL/HP/LENOVO EXHIBIT 1015

CEC 3 Feature Overview

CEC provides a number of recommended features designed to enhance the functionality and interoperability of devices within an HDMI system. This section gives an overview of these features.

CEC 3.1 End-User Features

One Touch Play - Allows a device to be played and become the active source with a single button press.

System Standby - Enables the user to switch all devices to standby with one button press.

One Touch Record - Offers a What You See Is What You Record (WYSIWYR) facility, meaning that whatever is shown on the TV screen is recorded on a selected recording device.

Deck Control - Enables a device to control (e.g. play, fast forward etc.) and interrogate a playback device (a deck).

Tuner Control - Allows a device to control the tuner of another device.

Device Menu Control - Enables a device to control the menu of another device by passing through user interface commands.

Remote Control Pass Through - Enables remote control commands to be passed through to other devices within the system.

CEC 3.2 Supporting Features

Device OSD Name Transfer - Enables devices to upload their preferred OSD name to the TV. The TV can then use this name in any menus associated with that device.

Device Power Status – Allows the current power status of a device to be discovered.

OSD Display - Enables a device to use the on-screen display of the TV to display text strings.

Routing Control - Allows the control of CEC Switches for streaming of a new source device.

System Information - Queries the system to determine device addresses and language.

Vendor Specific Commands - Allows a set of vendor-defined commands to be used between devices of that vendor.

CEC 4 Electrical Specification

The electrical specifications define CEC such that a maximum of 10 devices can interoperate in the worst-case scenario. In practice, many more may be expected to operate together as the worst case is highly improbable.

A device that implements CEC protocols, as described in this CEC supplement, and has enabled its CEC functionality, shall:

- Conform to Table 1 when it is powered-Off (e.g. power removed); or,
- Conform to Table 2 in all other power states. In these states, the device shall keep monitoring the CEC line for any messages addressing that device, including any messages that bring the device out of Standby, see CEC 14.1.3.

During the powered-Off state (e.g. power removed), the CEC line is not monitored.

CEC Table 1 CEC Electrical Specifications during the fully powered-Off state

Description	Value	Notes
Leakage current in powered-Off state	1.8μA max	1

CEC Table 2 CEC Electrical Specifications except during the fully powered-Off state

Description	Value	Notes
Maximum Output Voltage Logic '0'	+0.6V	
Minimum Output Voltage Logic '0'	0V	
Maximum Output Voltage Logic '1'	+3.63 V	
Minimum Output Voltage Logic '1'	2.5V	
High to Low Input Voltage Threshold Logic '0'	V_{cecin}('0')≥+0.8V	
Low to High Input Voltage Threshold Logic '1'	V_{cecin}('1') ≤ +2.0V	
Typical Input hysteresis	+0.4 V	2
Maximum rise time (10% to 90%)	250 μs	
Maximum fall time (90% to 10%)	50 μs	
Internal device pull-up:	27k ohms ± 5% or equivalent (eg a current source)	

The device shall remain within specification under the full-range of load conditions.

Notes:

1 This effectively requires that the internal pull-up circuit shall be disconnected from the CEC line when the device is off. For example, this can be implemented by connecting an isolating diode between the CEC input pin and the internal pull-up circuit, such that diode is reverse-biased in the off state with an external device pulling-up the CEC line.

2 For information, input hysteresis is normally supplied by the microprocessor input circuit: in this circumstance, external hysteresis circuitry is not needed.

DELL/HP/LENOVO EXHIBIT 1015
Page 537

CEC 5 Signaling and Bit Timings

All transactions on the CEC line consist of an initiator and one or more followers. The initiator is responsible for sending the message structure and the data. The follower is the recipient of any data and is responsible for setting any acknowledgement bits.

CEC 5.1 CEC Line Usage

A message is conveyed over the control signal line in a single frame; a frame is a self-contained unit consisting of a start bit followed by a number of data bits.

An initiator first has to test that the control signal line is free for use (described below). After that it generates a high to low transition on the CEC line, followed by a series of pulses comprising data bits whose starting point is defined by a high to low transition.

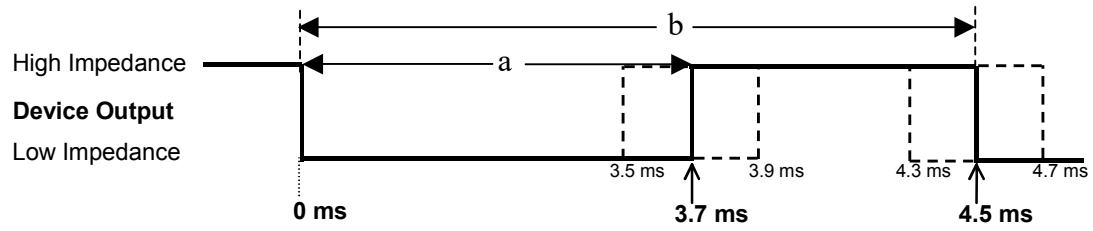
The initiator provides bit timing and bit leading edges. Only one initiator is allowed at any one time. A control signal line arbitration mechanism avoids conflict when more than one initiator begins transmitting at the same time.

CEC 5.2 Bit Timing

CEC 5.2.1 Start Bit Timing

The pulse format of the start bit is shown in CEC Figure 3. It is unique and identifies the start of a frame.

The start bit has to be validated by its low duration (a) and its total duration (b).

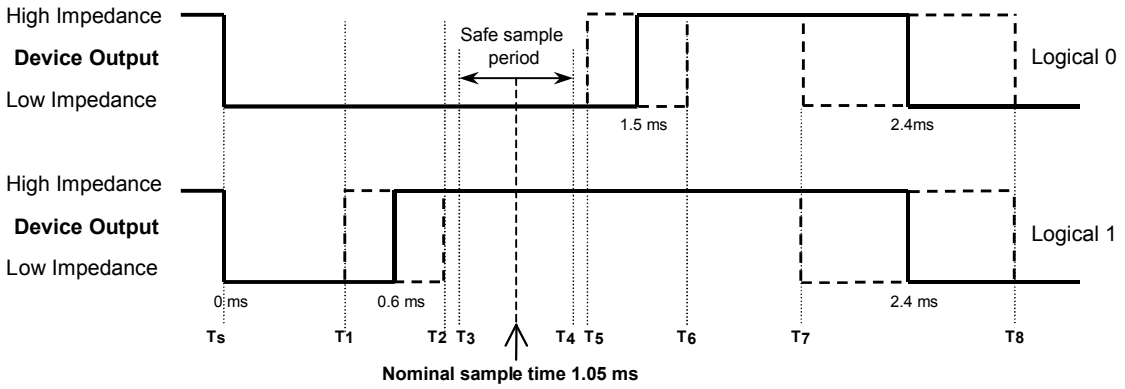


CEC Figure 3 Start bit pulse format showing minimum and maximum tolerances

CEC 5.2.2 Data Bit Timing

All remaining data bits in the frame, after the start bit, have consistent timing. There are, however, two types of bits; an initiator asserted bit and a follower asserted bit. All bits apart from the acknowledge bit are asserted by the initiator. CEC Figure 4 shows both logical 1 and logical 0 timing diagrams for an initiator asserted bit.

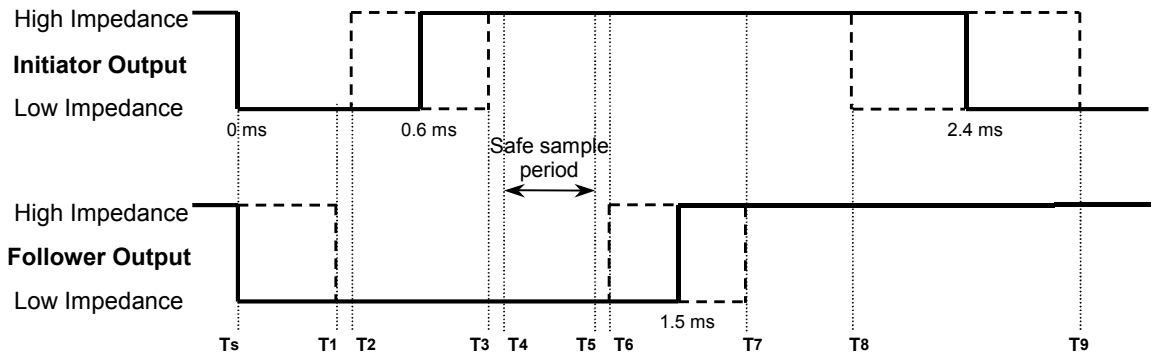
The high to low transition at the end of a data bit is the start of the next data bit and only occurs if there is a following data bit; after transmission of the final bit the CEC line remains high.



T_S	**0 ms**	The bit start event.
T_1	**0.4 ms**	The earliest time for a low - high transition when indicating a logical 1.
T_2	**0.8 ms**	The latest time for a low - high transition when indicating a logical 1.
T_3	**0.85 ms**	The earliest time it is safe to sample the signal line to determine its state.
T_4	**1.25 ms**	The latest time it is safe to sample the signal line to determine its state.
T_5	**1.3 ms**	The earliest time a device is permitted return to a high impedance state (logical 0).
T_6	**1.7 ms**	The latest time a device is permitted return to a high impedance state (logical 0).
T_7	**2.05 ms**	The earliest time for the start of a following bit.
	2.4ms	The nominal data bit period.
T_8	**2.75 ms**	The latest time for the start of a following bit.

CEC Figure 4 Timing diagrams for both bit states

CEC Figure 5 shows an example bit with both initiator and follower where the follower may assert the bit to a logical 0 to acknowledge a data block. The initiator outputs a logical 1, thus allowing the follower to change the CEC state by pulling the control line low for the duration of the safe sample period.



T_S	**0 ms**	The bit start event.
T_1	**0.35 ms**	The latest response time for a follower to go to the low impedance state.
T_2	**0.4 ms**	The earliest the initiator can return to high impedance when transmitting a logical 1.
T_3	**0.8 ms**	The latest the initiator can return to high impedance when transmitting a logical 1.
T_4	**0.85 ms**	The earliest time at which the bit state on the CEC line is valid for reading.
T_5	**1.25 ms**	The latest time at which the bit state on the CEC line is valid for reading.
T_6	**1.3 ms**	The earliest time the follower is permitted return to a high impedance state.
T_7	**1.7 ms**	The latest time the follower is permitted return to a high impedance state.
T_8	**2.05 ms**	The earliest time for the start of a following bit.
	2.4ms	The nominal data bit period.
T_9	**2.75 ms**	The latest time for the start of a following bit.

CEC Figure 5 Timing Diagram for Follower Asserted Bit (Logical 0)

CEC 6 Frame Description

The following table describes the complete CEC frame; the details of each block of the frame are given in the subsequent sections.

CEC Table 3 Frame Description

Name	Description	Value
Start	Special start 'bit'	N/A
Header Block	Source and destination addresses (see CEC Figure 7)	See CEC Table 5
Data Block 1 (opcode block)	Opcode (Optional)	See CEC Table 6 to CEC Table 18
Data Block 2…. (operand blocks)	Operand(s) specific to opcode (Optional, depending on opcode)	See CEC Table 6 and CEC Table 22

The maximum message size (opcode block plus operand blocks) is 16 * 10 bits

CEC 6.1 Header/Data Block description

All Data Blocks and Header Blocks are ten bits long and have the same basic structure, as shown in CEC Figure 6.

Header/Data Block									
7	6	5	4	3	2	1	0	-	-
Information bits								EOM	ACK

CEC Figure 6 Block Structure

The information bits are data, opcodes or addresses, dependent on context. The control bits, EOM and ACK, are always present and always have the same usage.

CEC 6.1.1 EOM (End of Message)

The EOM bit is used to indicate if this is the final block in the message.

A '0' bit specifies that one or more data blocks follow.

A '1' bit specifies that the message is complete.

In the event that a message contains additional data blocks after an EOM is indicated, the follower shall ignore the additional blocks.

CEC 6.1.2 ACK (Acknowledge)

The ACK bit is used by follower(s) to acknowledge the data or header block. It is always set to 1 by the initiator. It operates in one of two modes:

For messages addressed to a single device:

- A follower that reads its own address in the destination address field shall acknowledge with a '0' ACK bit.
- All other devices shall generate a '1' ACK bit (i.e. do nothing).
- A '0' read by the initiator therefore indicates a successful transmission of the data or header block.

For broadcast messages the sense of the ACK bit is inverted to allow for a single device to reject a message:

- All followers shall acknowledge with a '1' ACK bit.
- A '1' read by the initiator therefore indicates that no device has rejected the data or header block – the message transmission can therefore continue if required.
- A '0' read by the initiator therefore indicates that one or more devices have rejected the message.

CEC 6.1.3 Header Block Details

The header block consists of the source logical address field, the destination logical address field, the end of message bit (EOM) and the acknowledge bit (ACK) as shown in CEC Figure 7. The addresses for the devices are specified in CEC Table 5.

Header Block									
3	2	1	0	3	2	1	0	-	-
Initiator				Destination				EOM	ACK

CEC Figure 7 Header Block

The initiator logical address field is used to identify the initiator of the current frame. The logical address of the initiator is written in this field (see CEC 10.2). The field consists of bits one to four of the header block, most significant bit first.

The destination logical address field is used to identify the destination of the current frame. The logical address of the destination is written in this field (see CEC 10.2). A special address (0b1111) is used for broadcast messages. The field consists of bits five to eight of the header block, most significant bit first.

A message with the EOM bit set in the Header Block can be used to 'ping' other devices, to ascertain if they are active.

CEC 7 Reliable Communication Mechanisms

There are three mechanisms to provide a reliable communications medium for the transfer of frames:

- Frame re-transmissions increase the chance of a successful message transfer.
- Flow control ensures that communication only progresses as fast as the slowest follower.
- Frame validation.

Given these mechanisms and the active ACK method, a message transmitted and acknowledged should be assumed correctly received. A message that does not result in a <Feature Abort> can be assumed to have been acted upon. It is suggested that the receiving device can assume this after 1 second. Generally however, the <Feature Abort> will be received within around 100ms.

CEC 7.1 Frame Re-transmissions

A frame is considered lost and therefore may be re-transmitted under the following conditions:

- If a frame is not acknowledged in a directly addressed message.
- If a frame is negatively acknowledged in a broadcast message.
- If the initiator detects low impedance on the CEC line when it is transmitting high impedance and is not expecting a follower asserted bit.

Re-transmission can be attempted up to 5 times for a single message and shall be attempted at least once. The re-try shall be after a signal free time as described in CEC Table 4.

CEC 7.2 Flow Control

To provide flow control, a receiving device may negatively acknowledge any data or header block it is at present unable to process. A negative acknowledge will cause re-transmission by the initiator.

CEC 7.3 Frame Validation

A follower shall ignore a frame if the number of operands is less than the number specified for that opcode.

CEC 7.4 CEC Line Error Handling

It is the responsibility of all devices acting as followers to detect the existence of spurious pulses on the control signal line and notify all other devices (primarily the initiator) that a potential error has occurred.

An error is defined as a period between falling edges that is less than a minimum data bit period (i.e. too short to be a valid bit). Note that the start bit is excluded from CEC Line Error Handling.

Errors are notified by the follower generating a low bit period on the control signal line of 1.4-1.6 times the nominal data bit period. After such an error notification the original initiator should stop sending its current frame and re-try later.

CEC 8 Protocol Extensions

In order to allow for extensions to the protocol in future releases of the specification, the current opcodes and parameters can be extended by adding further parameters onto them. If an older CEC node receives a message with more operands than expected, it should ACK the additional operands and simply ignore them, thus allowing extensions to already existing commands.

For entirely new commands, new opcodes can be allocated.

For entirely new device types, new addresses may be allocated.

CEC 9 CEC Arbitration

Arbitration for the CEC line ensures collisions are spotted and a reliable message layer can be achieved.

All devices that wish to transmit a frame onto the CEC line have to ensure that it has been inactive for the signal free time, see CEC Table 4.

A device that has lost arbitration shall stop transmitting and become a follower. The device shall then wait for the CEC line to be inactive for the signal free time period as specified in CEC Table 4, before attempting to send another message.

CEC line arbitration commences with the leading edge of the start bit and continues until the end of the initiator address bits within the header block. During this period the initiator shall monitor the CEC line and if whilst in high impedance state it detects low impedance then it shall assume that it has lost the arbitration to a second initiator.

It should be noted that this process gives priority to the logical address with the most leading zeros and, ultimately, the TV.

CEC 9.1 Signal Free Time

Before attempting to transmit or re-transmit a frame, a device shall ensure that the CEC line has been inactive for a number of bit periods. This signal free time is defined as the time since the start of the final bit of the previous frame.

The length of the required signal free time depends on the current status of the control signal line and the initiating device. The different signal free times required are summarized in the following table:

CEC Table 4 Signal Free Time

Precondition	**Signal Free Time (nominal data bit periods)**
Present initiator wants to send another frame immediately after its previous frame	≥7 and <10
New initiator wants to send a frame	≥5 and <7
Previous attempt to send frame unsuccessful	≥3 and <5

CEC 9.2 Message Time Constraints

There are certain time constraints for messages that should be obeyed at application level. These are a desired maximum response time of 200ms and a required maximum response time of 1 second.

CEC 10 Device Connectivity and Addressing

CEC is a protocol based on a bus system and therefore cannot alone ascertain the physical connectivity of the network. The mechanism defined in section 8.7 uses DDC to allocate physical addresses to devices in the network.

All CEC devices therefore have both a physical and logical address, whereas non-CEC devices only have a physical address.

CEC 10.1 Physical Address Discovery

The algorithm defined in 8.7.3 is used to allocate the physical address of each device.

Whenever a new physical address (other than F.F.F.F) is discovered, a CEC device shall:

- allocate the logical address (see CEC 10.2.1)
- report the association between its logical and physical addresses by broadcasting <Report Physical Address>.

This process allows any node to create a map of physical connections to logical addresses.

CEC 10.2 Logical Addressing

Each device appearing on the control signal line has a unique logical address. This address defines a device type as well as being a unique identifier. These are specified in CEC Table 5.

If a physical device contains the functions of more than one logical device then it should take the logical addresses for each of those logical devices. It is allowed for a device to declare the functionality of another device by using a different logical address. For example a recordable DVD device may take the address 4 or 8 to expose only the functionality of a standard DVD device.

CEC Table 5 Logical Addresses

Address	Device
0	TV
1	Recording Device 1
2	Recording Device 2
3	STB1
4	DVD1
5	Audio System
6	STB2
7	STB3
8	DVD2
9	Recording Device 3
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Free Use
15	Unregistered (as initiator address) Broadcast (as destination address)

CEC 10.2.1 Logical Address Allocation

Note that a logical address should only be allocated when a device has a valid physical address (i.e. not F.F.F.F), at all other times a device should take the 'Unregistered' logical address (15).

Only the device at physical address 0.0.0.0 may take logical address TV (0). A TV at any other physical address shall take the 'Free Use' (14) address. If address 14 is already allocated it shall take the 'Unregistered' address (15).

Reserved addresses shall not be used at present and are reserved for future extensions to this specification.

Where more than one possible logical address is available for the given device type (e.g. STB1, STB2, etc.), an address allocation procedure shall be carried out by a newly connected device. The device takes the first allocated address and sends a polling message to the same address (e.g. STB1 → STB1). If the polling message is not acknowledged, then the device stops the procedure and retains that address.

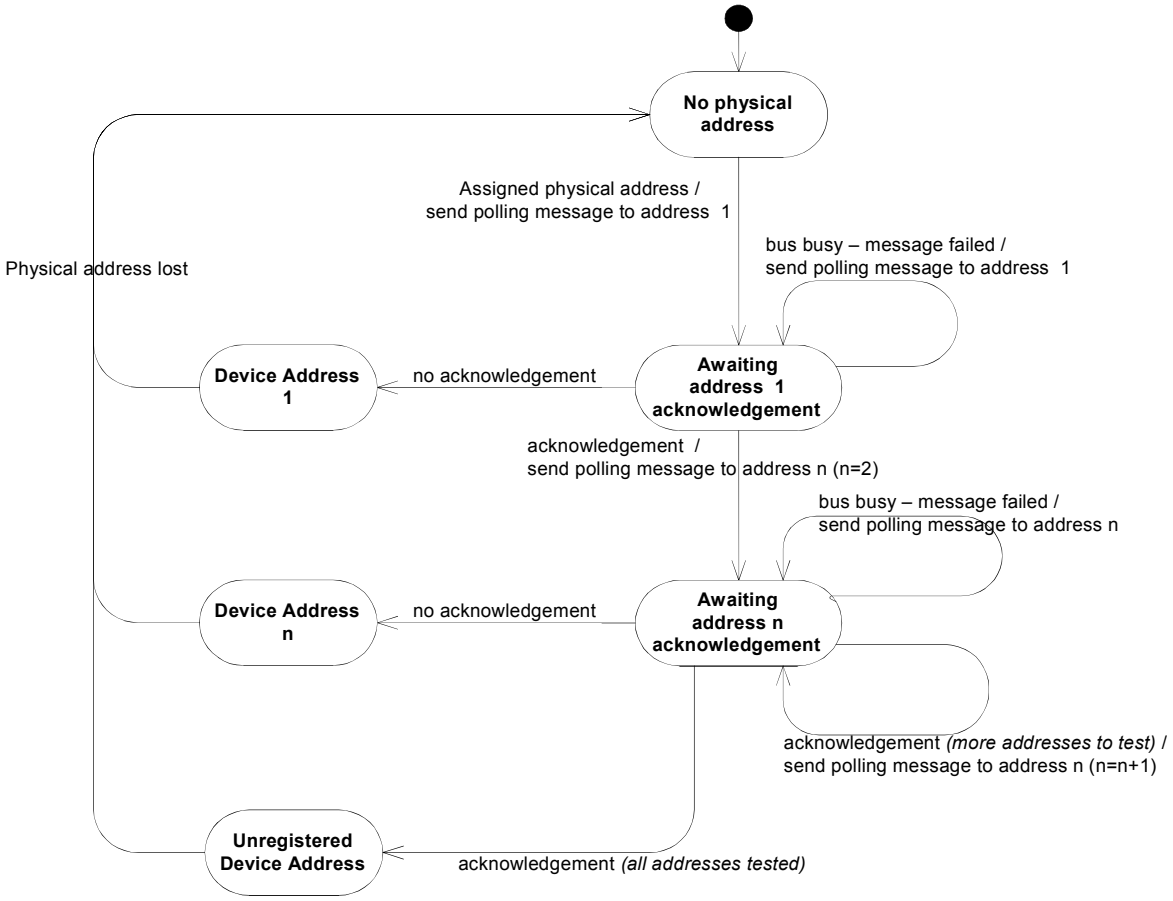
If the first address is acknowledged, then the device takes the next address and repeats the process (e.g. STB2 → STB2). Again, if the message is not acknowledged, the device keeps that address.

This procedure continues until all possible 'type specific' addresses have been checked; if no 'type specific' addresses are available the device should take the unregistered address (15).

A device may lose its logical address when it is disconnected or switched off. However, it may remember its previous logical address, so that the next time it is reconnected, it can begin the polling process at its

previous logical address and try each other allowable logical address in sequence before taking the unregistered address. For example if an STB that was previously allocated address STB2 is reconnected, it would poll STB2, STB3 and STB1 before taking the unregistered address.

If a device loses its physical address at any time (e.g. it is unplugged) then its logical address shall be set to unregistered (15).



CEC Figure 8 Logical Address Allocation

CEC 11 Switch Requirements

One of the major uses of the physical address identification is to allow a switch to be controlled in order to enable a specific device to stream to the TV. This places requirements on both CEC and non-CEC compliant switches.

All switches will consist of a single switched TMDS connection, and a fully wired CEC connection to each source device. There are significant differences between a CEC Switch and a non-CEC Switch, which are identified below.

CEC 11.1 CEC Switch

A CEC Switch allocates a unique child_address for every connection below the switch. This means that any device connected to the switch will always have a valid physical address (assuming the switch itself has a valid physical address). Therefore, any device below the switch may take a logical address and can react to CEC messages in a normal way. The switch is effectively transparent and will enable all standard CEC communications in its connected source devices.

For CEC compliant switches, there is a requirement to react on <Active Source> and <Set Stream Path> messages. Both of these messages require the switch to change the connected source device according to the physical AV stream path indicated by the CEC message. These mechanisms allow a source device to configure the switches between itself and the TV to ensure that its output is displayed, or for the TV to specifically receive the output from a given device.

A CEC compliant device that is only a switch does not need to take a logical address; it uses the unregistered address for any communications.

It is possible that a user may change a CEC Switch manually. In this instance a CEC Switch shall send a <Routing Change> message to inform other devices about the change.

CEC 11.2 Non-CEC Switch

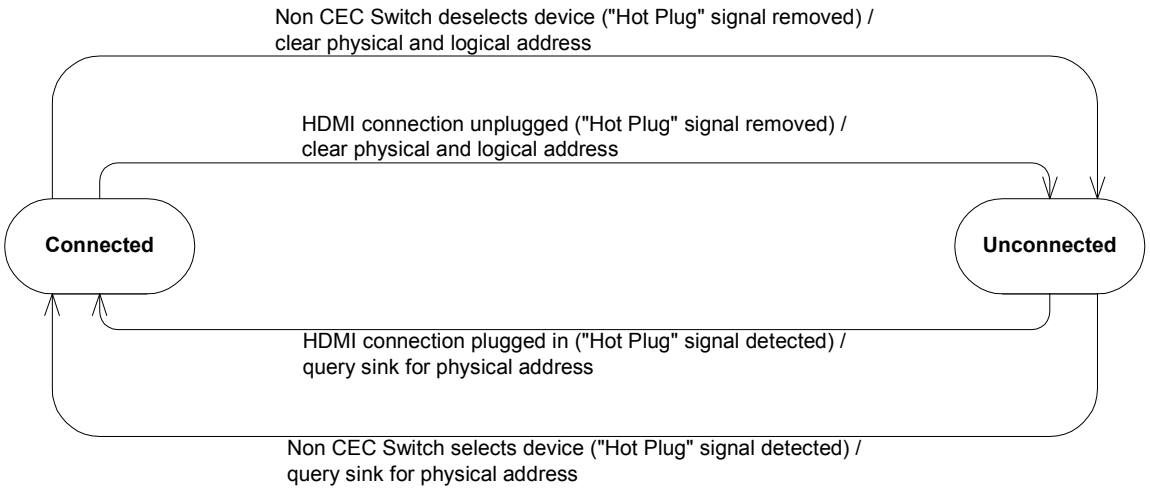
There are two types of non-CEC switches, those which have only one EDID for all source devices (or simply reflect the sink EDID), and those which have a separate EDID for all source devices. The rules for the operation of these two types of switch are different:

CEC 11.2.1 Switches with One EDID

A non-CEC-compliant switch may have a single child_address, which is always occupied by the currently switched device. Any other connected devices will have no hot plug signal and will therefore have an unallocated physical address (and can use only the unregistered logical address). These devices will, however, still see CEC messages as they will be connected to the CEC line and they may react to some broadcast messages in the normal way (e.g. standby).

When a switch de-selects a device, that device will detect the removal of the ‘hot plug’ signal to indicate that its physical AV connection has been removed. It should immediately clear its physical and logical addresses. Each source device below the switch will detect the removal of the ‘hot plug’ signal to indicate they are no longer on the active AV Path and clear their addresses accordingly.

When a switch selects a device, that device will detect the ‘hot plug’ signal. It can then obtain a valid physical address from its sink and subsequently a logical address. The device should activate the hot plug signal to its source (child) devices (if any) to indicate that they should now request a physical address.



CEC Figure 9 Behavior when beneath a one EDID non-CEC Switch

CEC 11.2.2 Switches with Multiple EDIDs

These should operate as CEC switches except that they do not send messages on, or monitor, the CEC line.

CEC 12 High Level Protocol

As described in the previous sections, messages consist of an opcode and a number of parameters. This is the high level protocol.

This protocol can be described best by detailing the messages and the data types used for the parameters. These are detailed in CEC Table 6.

Although these tables explain the majority of the high level protocol, there are some special situations that require further explanation. These are given in the following sections.

CEC 12.1 Source Declaration

Before a device becomes a source, it should issue an <Active Source> message to declare its intention. Thus any presently active source can and should act appropriately.

CEC 12.2 Protocol General Rules

A message that is defined as being valid only when directly addressed shall be ignored if received as a broadcast message.

A message that is defined as being valid only when broadcast shall be ignored if received as a directly addressed message.

All numbers greater than one byte are transmitted as bytes in big endian format.

All bit sequences are sent most significant bit first.

A follower shall respond to a message coming from any valid logical address from 0 to 14 unless otherwise stated.

A follower shall ignore a message coming from address 15 (unregistered), unless:

- that message invokes a broadcast response (e.g. <Get Menu Language >), or,
- the message has been sent by a CEC switch (a <Routing Change> or <Routing Information> message), or,
- the message is <standby>.

CEC 12.3 Feature Abort

All devices shall support the message <Feature Abort>. It is used to allow devices to indicate if they do not support an op code that has been sent to them, if it is unable to deal with the message at present, or if there was something wrong with the transmitted frame at the high-level protocol layer.

Feature abort has two parameters, the opcode and a reason for its rejection of the frame.

The reaction to a faulty message by the follower depends on if the message was directed or broadcast:

For a broadcast message:

- A follower that receives a broadcast message which it does not support, ignores the received message, and does not send a <Feature Abort>.

For a directly addressed message:

- <Feature abort> is used as a response to any failure.

If an initiator wishes to attempt retransmission after receiving a <Feature Abort> it is suggested that it waits for 200ms. This will allow time for the follower to recover from the state that caused the initial <Feature Abort> message.

- <Feature Abort> is also used as a response to the <Abort> message during testing, see CEC 12.4

CEC 12.4 Abort

The <Abort> message shall be implemented as a Follower in all devices except CEC switches and is used during testing only. It shall be directly addressed to a specific device, which shall respond with a <Feature Abort> message. In this instance, any valid [Abort Reason] operand may be returned.

- A device shall ignore an <Abort> message which is broadcast.

CEC 13 CEC Features Description

This section describes the message transfer and additional details for a number of common features enabled by CEC. Note that where a feature is supported, all messages within that feature should be implemented.

CEC 13.1 One Touch Play

CEC 13.1.1 Messages

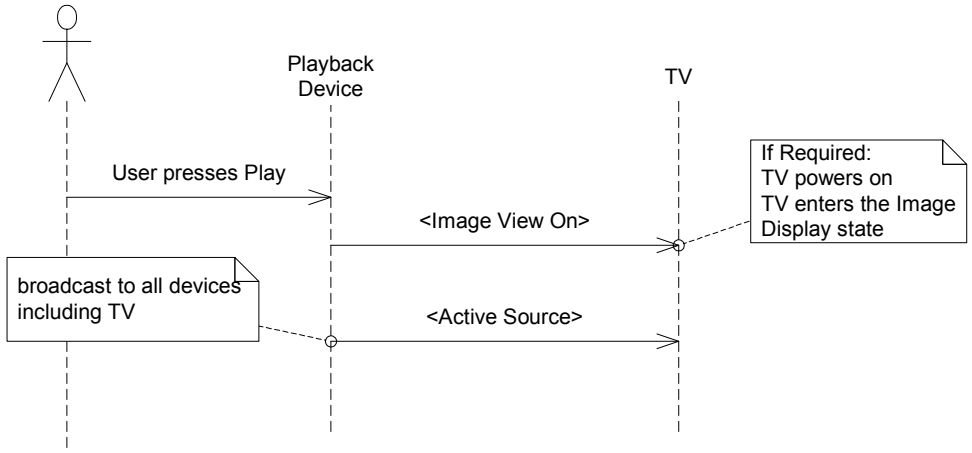
The following messages are used for the One Touch Play feature:

<Active Source>, <Image View On>, <Text View On>

CEC 13.1.2 Feature Description

The One Touch Play feature allows a device to be played and become the active source with a single button press.

A device shall send the message <Image View On> to the TV to indicate that its output should be displayed on the screen. If the TV is in a Text Display state (e.g. Teletext) it should switch to the Image Display state. If a menu is being displayed on the TV it shall remain on screen.



CEC Figure 10 A typical scenario illustrating the One Touch Play feature

A device may alternatively send the message <Text View On>. This message has the same functionality as <Image View On> with the addition that any menus that the TV is displaying shall be removed.

Whenever a device becomes the new source device it should broadcast an <Active Source> message. The current active source device should then take appropriate action, for example, it may pause if it is playing media.

Note: there is a special case when a TV switches to its internal tuner or to another non-HDMI source (eg Y/C, or a SCART socket on European market sets). In this case, it is the TV which broadcasts the <Active Source> message with address 0.0.0.0.

Note that it is mandatory for a source to implement at least one of <Image View On> or <Text View On>.

CEC 13.2 Routing Control

CEC 13.2.1 Messages

The following messages are used for the Routing Control feature:

<Active Source>, <Request Active Source>, <Set Stream Path>, <Routing Change>, <Routing Information>

CEC 13.2.2 Feature Description

This feature is used to control the routing of the HDMI network, by controlling CEC Switches. In general whenever a device starts being streamed to the TV it should send an <Active Source> message (see One-Touch Play in section CEC 13.1).

On receiving an <Active Source> message, CEC Switches between the device and the TV shall come out of standby (if necessary) and switch (if required) to ensure the device at the physical address specified is on the active path.

If a device (other than a CEC switch) is currently the active source, then it shall lose its active source status on receiving an <Active Source> message from another device and should act accordingly, for example it may pause if it is playing media.

When a device comes out of standby or a (mains) off state, it may broadcast a <Request Active Source> message to discover if any other device is currently acting as the active source. On receiving a <Request Active Source> message, the active source device shall respond by broadcasting an <Active Source> message. A particular instance of this rule is when a TV comes out of standby some time after its source device(s). In this case, if the TV has multiple HDMI connectors, it will not know which is the correct connector for the currently active source (because it was in standby when that device sent its <Active Source> message). Here, the <Request Active Source> and the corresponding <Active Source> response are needed to identify the relevant connector.

The user may select a device to view via the TV user interface. In contrast to the <Active Source> message (which is sent by the current active source to the TV), the <Set Stream Path> is sent by the TV to the source device to request it to broadcast its path using an < Active Source> message. In this case, the TV should broadcast a <Set Stream Path> message with the physical address of the device it wishes to display as a parameter. Any CEC Switches between the device and TV shall switch (if required) to ensure the device is on the active AV path. This feature also ensures that non-CEC-compliant devices in the network can be switched to, if for instance they have been manually set up in the TV menu. A CEC device at the location specified by the <Set Stream Path> message shall broadcast an <Active Source> message and begin streaming its output.

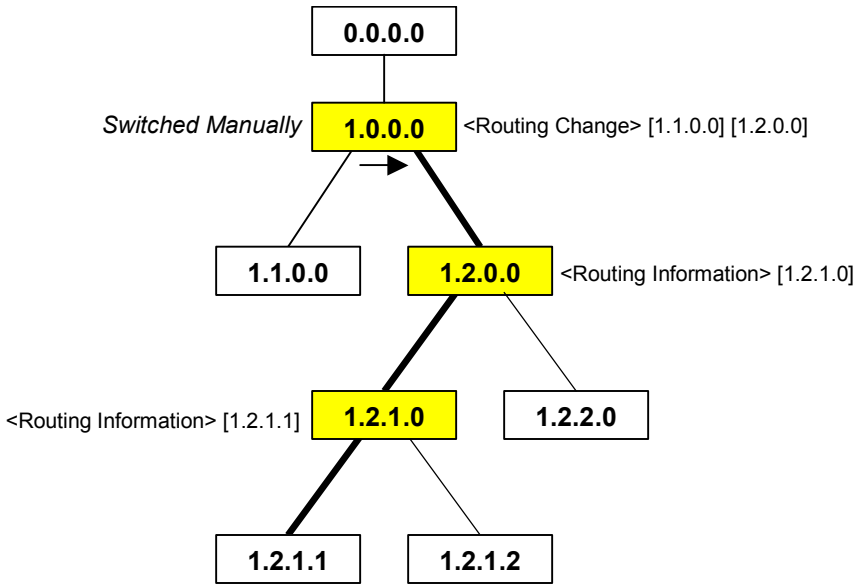
Note: there is a special case when a TV switches to its internal tuner or to another non-HDMI source (eg Y/C, or a SCART socket on European market sets). In this case, it is the TV which broadcasts the <Active Source> message with address 0.0.0.0.

In the case that the user manually switches a CEC Switch it should broadcast a <Routing Change> message. This will inform all devices in the network that the current active route below the switch has changed. This enables, for instance, a device that has been deselected by a switch to pause if it is playing media.

If a CEC Switch is at the new position indicated by the <Routing Change> message then it shall broadcast a <Routing Information> message with the physical address of its current active path. If a CEC Switch receives a <Routing Information> message then it shall broadcast a <Routing Information> message to indicate it's current active path. In this way the TV is aware of which device is on the current active path after the switch occurs. If the TV detects that the active source device has been de-selected by changing the switch it may either switch to an internal digital service or send a <Set Stream Path> message to the device at the new location to indicate that it should become the new active source.

The TV shall wait for a minimum of 7 nominal data bit periods before reacting to a <Routing Change> or <Routing Information> message to allow CEC switches to relay any <Routing Information> messages that are required.

The following diagram shows an example of the message flow when a user manually switches a CEC Switch. (CEC Switches are shown filled).



CEC Figure 11 Example message flow, when a CEC Switch is manually switched

CEC 13.3 System Standby

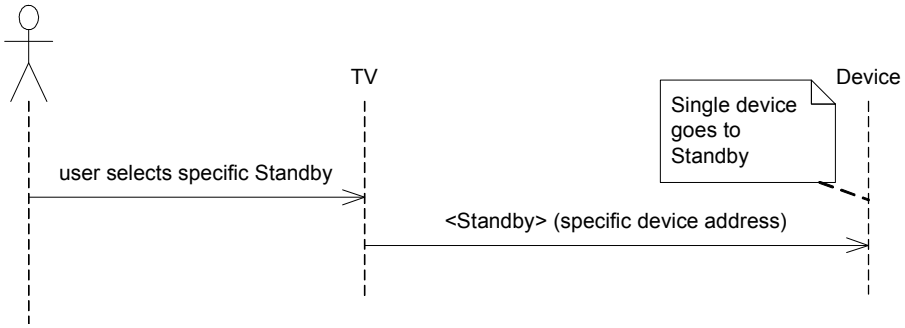
CEC 13.3.1 Messages

The following message is used for the System Standby feature:

<Standby>

CEC 13.3.2 Feature Description

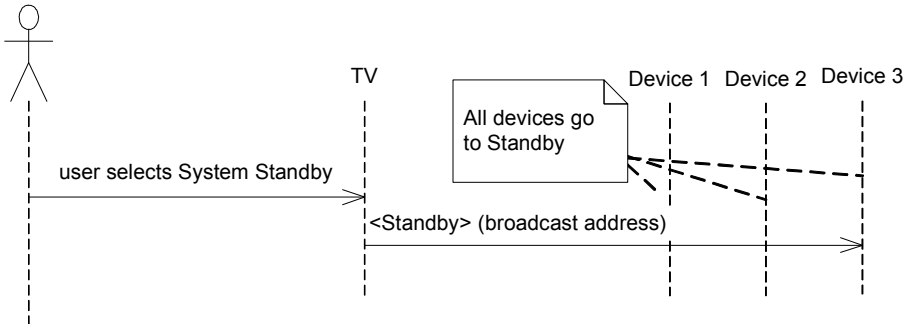
The message <Standby> can be used to switch one or more CEC devices to standby. A typical scenario where the user sets a specific device to standby is shown below:



CEC Figure 12 A typical scenario for the Standby feature to a specific device

A device can switch another single device into standby by sending the message <Standby> as a directly addressed message to it. It should only switch a single device into standby if it previously activated that device (brought it out of standby).

The whole system may be set to standby by broadcasting the <Standby> message. It is manufacturer dependent on how to differentiate between standby for a single device, e.g. a STB, and System Standby (broadcast to the whole system).



CEC Figure 13 A typical scenario for the broadcast (system) Standby feature

Note that whilst it is possible for a recorder to ignore a <Standby> message (see “Exceptions” below), it is not possible to identify the source for a recording device, e.g. a STB to a separate recorder. This implies that:

- With a specific <Standby> message, if the TV has brought a device (eg an STB) out of standby and it then sends a specific <Standby> to that STB (as a result of user interaction or when the TV goes to standby), this may upset a recording. It is therefore a manufacturer’s decision whether to send a specific <Standby> under these circumstances.
- The broadcast <Standby> message may cause the source of a recording to go to standby. Therefore, if the device is capable of sending a broadcast <Standby> message, then it should also provide a means for the user to select between the system (broadcast) <Standby> or the specific <Standby> message.
- It is also the manufacturer's decision if a source device goes into standby when it receives a system (broadcast) <Standby> message.

Devices can ignore <Standby> messages if they are in a state where going into standby is not sensible. For example if the device is actively involved in a recording it should ignore any requests to go into standby. It is up to the device implementer how ‘sensible’ is defined.

CEC 13.4 One Touch Record

CEC 13.4.1 Messages

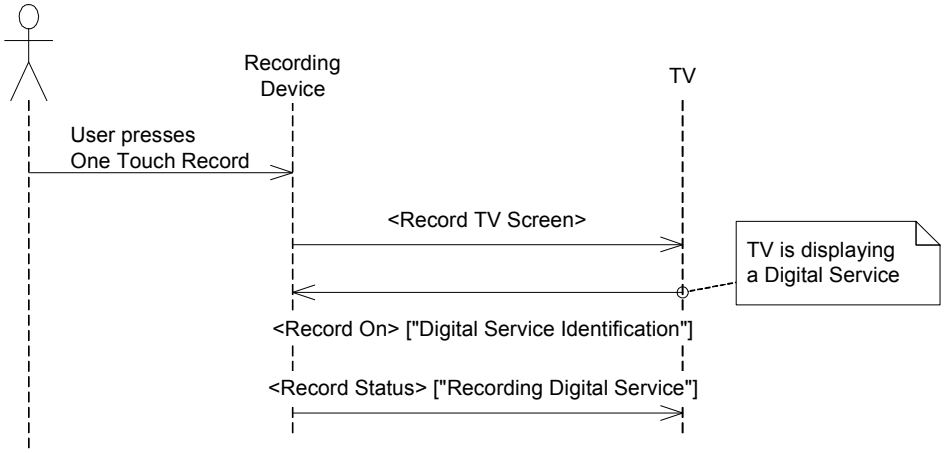
The following messages are used for the One Touch Record feature:

<Record Off>, <Record On>, <Record Status>, <Record TV Screen>

CEC 13.4.2 Feature Description

This feature allows the user to easily start a recording of the source that is being displayed on the TV, just by selecting a recording device and giving the record command. It is not always possible to carry out a One Touch Record as it depends on what source is currently being displayed. It is primarily used for the instant recording of a tuner preset, or the recording of another device (e.g. Camcorder) connected externally to the recording device

Either the TV or the recording device may initiate the One Touch Record Feature, for example by selecting a menu option on the TV or by pressing record on the recording device.



CEC Figure 14 A typical scenario for the One Touch Record feature

In the event of the recording device initiating the feature it will send a <Record TV Screen> message to the TV. On receipt of the <Record TV Screen> message by the TV, or if the user initiates the One Touch Record feature via the TV, the TV shall react as follows:

- It the TV is currently displaying an internal digital tuner service, it shall respond with a <Record On> [“Digital Service”] [Digital Service Identification] message.
- If the Recording Device is the current active source device, then the TV shall respond with a <Record On> [“Own source”] message.
- If the TV is currently displaying some other source, it shall respond with a <Feature Abort> [“Cannot provide source”] message, or do nothing if initiated via the TV.

On receipt of a <Record On> message the recording device shall act as follows:

- If [“Digital Service”] is indicated and the device can record the digital service using the Digital Service Information that was sent, the device shall change to that digital service and start recording. If the device cannot support the required digital service, then it should respond with a <Record Status> [“No recording –unable to record Digital Service”]. In this latter case, the TV should inform the user that the recording has failed, with the reason, or take other appropriate action.

- If [“Own source”] is indicated, then it shall attempt to record whatever it is currently displaying, e.g. an external connection such as a camcorder or the digital service it is currently tuned to.

The recording device shall respond with the message <Record Status> to indicate if recording has begun, or a reason why recording has failed. If the TV initiated the recording, it may stop it at any time by sending a <Record Off> message. The recording device should then stop recording immediately.

The TV should ignore a <Record TV Screen> message that comes from a non-recording device address, however it shall accept the message from a ‘Reserved’ address (a future device type).

CEC 13.5 System Information

CEC 13.5.1 Messages

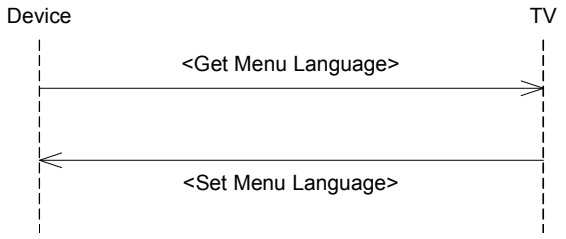
The following messages are used for the System Information feature:

<Get Menu Language>, <Give Physical Address>, <Polling Message>, <Report Physical Address>, <Set Menu Language>.

CEC 13.5.2 Feature Description

This feature allows devices to automatically use the same language settings as the TV and also for a TV to discover the current language when it is being installed.

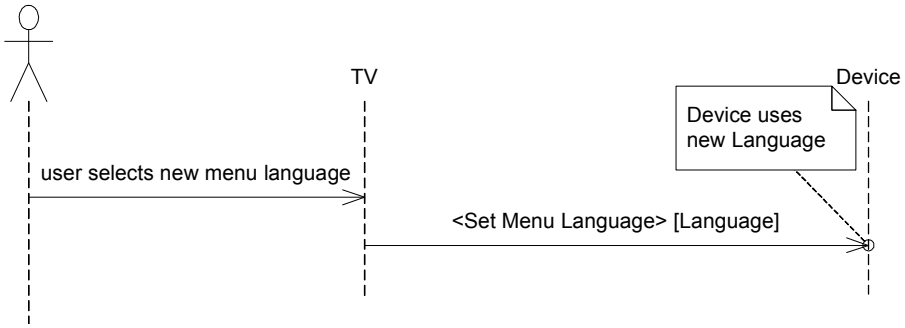
When a source device is powered on, it should send a <Get Menu Language> message to the TV. The TV shall then respond as shown below with a <Set Menu Language> message.



CEC Figure 15 Message exchange when getting the TV’s menu Language

When the user changes a menu language setting on the TV, it shall send a <Set Menu Language> message containing the currently selected menu [Language], as shown below.

DELL/HP/LENOVO EXHIBIT 1015



CEC Figure 16 A typical scenario when a menu language setting within the TV is modified

On receipt of the <Set Menu Language> message, the device shall attempt to use the newly selected [Language] for Menus and OSDs.

Note that a device might receive a <Set Menu Language> message even when the language has not been changed. A device shall ignore any of the above messages that come from an initiator address other than 0 (the TV).

During the installation of a TV, the <Get Menu Language> message may be sent by the TV to another device to discover what language has been set on that device.

CEC 13.5.3 Additional Information

The <Polling Message> is used to detect the presence or absence of a device within the system. It is used for allocating logical addresses as detailed in CEC 10.2.1.

The <Report Physical Address> message is used by a device to broadcast its physical address to all other devices in the system. By using the logical and physical addresses, any other device is able to derive the physical connectivity of the network. A device may request the physical address of another device by sending a directly addressed <Give Physical Address> message to it.

CEC 13.6 Deck Control

CEC 13.6.1 Messages

The following messages are used for the Deck Control feature:

<Deck Status>, <Give Deck Status>, <Deck Control>, <Play>

CEC 13.6.2 Feature Description

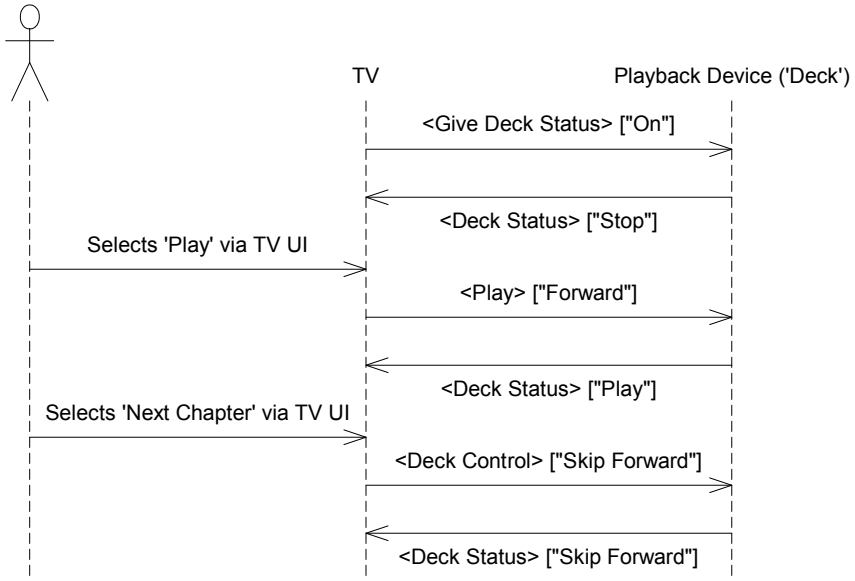
This feature allows a playback device (a deck or disc player or recorder) to be controlled by another device (e.g. the TV). Messages are also provided to allow a device to find out the status of the Deck; this allows, for example, a TV to keep its user interface synchronized with the status of the Deck.

A device may query the status of a deck with the <Give Deck Status> command. The deck should respond with a <Deck Status> message.

A device may control a Deck with the <Play> and <Deck Control> messages. These messages may be initiated after a user command. The Deck shall act upon the command that it receives within the messages <Play> and <Deck Control>. It is the equivalent of the user selecting the command local to the Deck. If the

deck cannot carry out the command (e.g. it has no media when trying to play) it should respond with a <Feature Abort> [“Not in correct mode to respond”] message.

If the deck is in standby and receives a <Deck Control> [“Eject”] or <Play> [“Play Forward”] message, it should power on and act on the message. It is up to the manufacturer to decide if the device should power on when receiving any other <Deck Control> or <Play> messages.



CEC Figure 17 A typical scenario for the Deck Control feature

Note that it is mandatory for a Deck to support <Play>[“Forward”] and <Deck Control>[“Stop”] only. The support of <Play> and <Deck Control> with other operands is optional.

CEC 13.7 Tuner Control

CEC 13.7.1 Messages

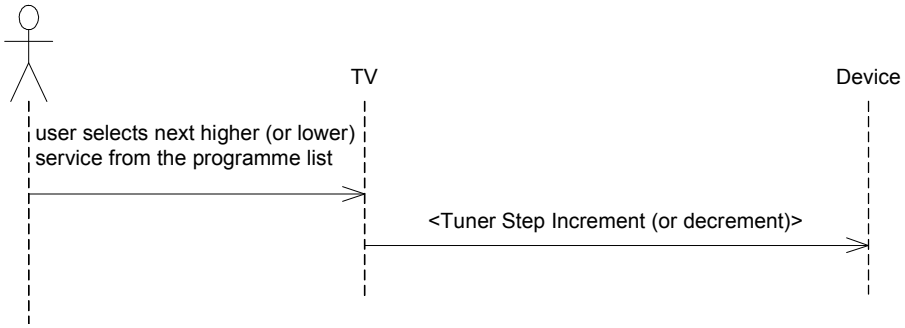
The following message are used for the Tuner Control feature:

<Give Tuner Device Status>, <Select Digital Service>. <Tuner Step Decrement>, <Tuner Step Increment>, <Tuner Device Status>

CEC 13.7.2 Feature Description

This feature allows a device (e.g. the TV) to control another CEC device’s tuner.

DELL/HP/LENOVO EXHIBIT 1015
Page 560



CEC Figure 18 A typical scenario for selecting a new Digital Service

A device can select a digital service on a tuner device by sending the <Select Digital Service> message. The tuner device shall then attempt to tune to that digital service and stream its output on the HDMI connection. If the specified digital service is not supported on the device then it should send a <Feature Abort> ["Invalid operand"] message. If the tuner device cannot select that digital service (e.g. if it is recording), it should respond with a <Feature Abort> ["Refused"] message.

A device can also control a CEC Device's tuner using the <Tuner Step Increment> and <Tuner Step Decrement> messages. If a device receives the <Tuner Step Increment> or <Tuner Step Decrement> message then it should select the next highest or next lowest digital service in its service list. The tuner device can interpret the messages as it chooses, for example, it may only cycle through channels included in the user's list of favorites.

A device may query the status of a tuner device by sending a <Give Tuner Device Status> message. The tuner device shall respond by sending a <Tuner Device Status> message indicating if it is currently displaying its tuner and the digital service that is currently selected.

CEC 13.8 Vendor Specific Commands

CEC 13.8.1 Messages

The following messages are used for the Vendor Specific Commands feature:

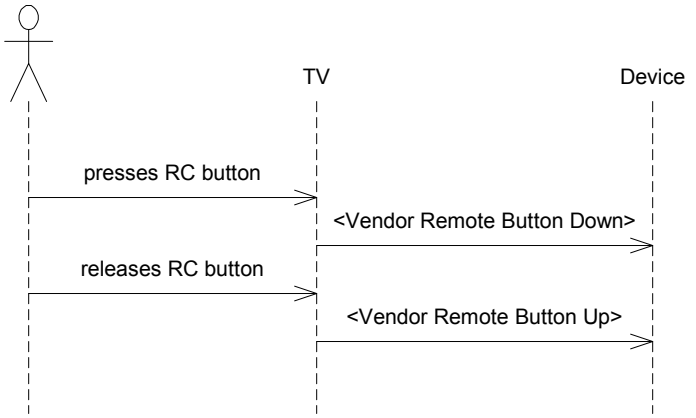
<Device Vendor ID>, <Give Device Vendor ID>, <Vendor Command>, <Vendor Remote Button Down>, <Vendor Remote Button Up>

CEC 13.8.2 Feature Description

This feature allows a set of vendor specific commands to be used to communicate between devices of a single vendor.

A device that supports vendor specific commands shall store a Vendor ID. A device shall broadcast a <Device Vendor ID> message after a successful initialization and address allocation to inform all other devices of its vendor ID. A device may request the Vendor ID of another device by sending a <Give Device Vendor ID> message to it. The follower shall respond by broadcasting a <Device Vendor ID> message. In this way any device can determine the vendor of another device. A device shall only attempt to transmit a vendor specific command to another device that it knows has the same vendor ID. In this way a device receiving a vendor specific command can always assume that it came from a device of the same vendor and respond accordingly.

It is possible to send vendor specific remote control commands using the <Vendor Remote Button Down> and <Vendor Remote Button Up> messages.



CEC Figure 19 The messages sent in the Vendor Specific Commands feature

In addition it is possible to send other (non remote control key) vendor specific messages using the <Vendor Command> message. The message parameter(s) can be used to communicate any additional (vendor defined) messages and data.

CEC 13.9 OSD Status Display

CEC 13.9.1 Messages

The following message is used for the OSD Status Display feature:

<Set OSD String>

CEC 13.9.2 Feature Description

This feature allows a device to transfer a text string to the TV for On Screen Display. The <Set OSD String> message is used to transfer the text string to the TV.

A text string may be displayed for a default period (i.e. 5 seconds) or until explicitly cleared. In the latter case the device should send another <Set OSD String> message to clear the text when it is appropriate.

The TV should display the whole string unless it is in an unsuitable state, in which case it should generate a <Feature Abort> message.

If a new <Set OSD String> message is received when an OSD String is already being displayed, it should overwrite the existing string. OSD Strings generated locally within the TV may also overwrite any messages sent via the <Set OSD String> message.

CEC 13.10 Device OSD Name Transfer

CEC 13.10.1 Messages

The following messages are used for the Device OSD Name Transfer feature:

<Give OSD Name>, <Set OSD Name>

CEC 13.10.2 Feature Description

This feature is used to request the preferred name of a device to be used in any on screen display (e.g. menus), which reference that device. A device (e.g. the TV) may request another devices name by sending a directly addressed <Give OSD Name> message to it. If the device supports this feature it shall respond with a <Set OSD Name> message. The devices name should then be stored and used in any future on screen references to it.

A TV shall send a <Give OSD Name> message whenever it discovers a new device that has been connected.

CEC 13.11 Device Menu Control

CEC 13.11.1 Messages

The following messages are used for the Device Menu Control feature:

<User Control Pressed>, <User Control Released>, <Menu Request>, <Menu Status>

CEC 13.11.2 Feature Description

This feature allows device menus to be controlled via the TV remote control as if it was using its own remote control and allow the TV to be aware when another device has a menu on its display.

A device shall indicate that it is displaying a menu by sending a <Menu Status> [“Activated”] message to the TV. If the device leaves the menu it shall send a <Menu Status> [“Deactivated”] message to the TV. The TV should then handle incoming remote control commands internally (as it would normally).

The message <User Control Pressed> can be used to send incoming Remote Control commands from the TV to a device that it is displaying a menu. The <User Control Released> message should be sent on release of the RC button. If a device fails to acknowledge any <User Control Pressed> or <User Control Released> message when in the providing menu state, the TV shall assume that it has been removed from the system and act accordingly. For more information on <User Control> see the Remote Control Pass Through feature description (CEC 13.12).

The TV may initiate a device’s menu by sending a <Menu Request> [“Activate”] command. It may subsequently remove the menu by sending a <Menu Request> [“Deactivate”] message. The TV may also query a devices menu status by sending a <Menu Request> [“Query”]. The menu device shall always respond with a <Menu Status> command when it receives a <Menu Request>.

A new active source device shall send a <Menu Status> [“Activated”] message to the TV if it is displaying a menu. The TV shall assume that a new active source is not in a menu unless it receives this message after the <Active Source> message. The TV shall ignore a <Menu Status> message coming from a device that is not the current active source. A source device shall only send <Menu Status> commands when it is the current active source.

CEC 13.12 Remote Control Pass Through

CEC 13.12.1 Messages

The following messages are used for the Remote Control Pass Through feature:

<User Control Pressed>, <User Control Released>

CEC 13.12.2 Feature Description

This feature is used to pass remote control commands received by one device (typically the TV) through to another device in the network. This feature will typically be used in situations where a TV offers a remote control with additional modes for controlling other devices within the system. The TV will receive the RC command and pass the command through to the appropriate device within the system.

The initiator shall decide on which device the remote command is for and send a directly addressed <User Control Pressed> message to it using the table of standard remote control pass through commands given in CEC Table 23. In a system where there is more than one of a particular device type present, the initiator should decide (as locally specified) a default device to pass remote control commands to.

The initiator will send a <User Control Pressed> message when the remote control button is pressed. When the button is released a <User Control Released> message should be sent by the initiator. The initiator should not send repeated <User Control Pressed> messages for the same button press.

The initiator may send further <User Control Pressed> messages without interleaving <User Control Released> messages if a new button press occurs quickly after a button release.

A device that has initiated a <User Control Pressed> message shall ensure that it sends a <User Control Released> message before going into standby. In the event that the initiator of the message is powered off/disconnected before sending a <User Control Released> message, the follower will never receive the <User Control Released> message.

This method should not be used for sending commands other than true remote control pass through to another device as the actions taken by the other device are not defined in a consistent way.

CEC 13.13 Give Device Power Status

CEC 13.13.1 Messages

The following messages are used for the Give Device Power Status feature:

<Give Device Power Status>, <Report Power Status>

CEC 13.13.2 Feature Description

Several messages, such as <Image View On> and <Play>, bring another device out of standby. The <Give Device Power Status> message is used to determine the current power status of a target device. The target device responds with a <Report Power Status> message containing the Power Status operand.

Some devices, such as TVs, may take some time before they have fully transitioned to the On state. A requesting device may poll the target device to determine when that device is fully On. In this case, the requesting device shall not send a <Give Device Power Status> message more frequently than once every 0.5 seconds. It is not recommended that a requesting device polls another device until the first device has transitioned to a stable state.

CEC 14 Device States

This section shows how CEC messages can change the states of a device.

CEC 14.1.1 Device States

The following is a list of states that each device type can be in. Each device should be in one and only one state for each line shown below.

All Devices: On, Standby, Off

TV: Image Display, Menu Display, Text Display
Device Menu Active, Device Menu Inactive

Recording Device: Recording, Not Recording

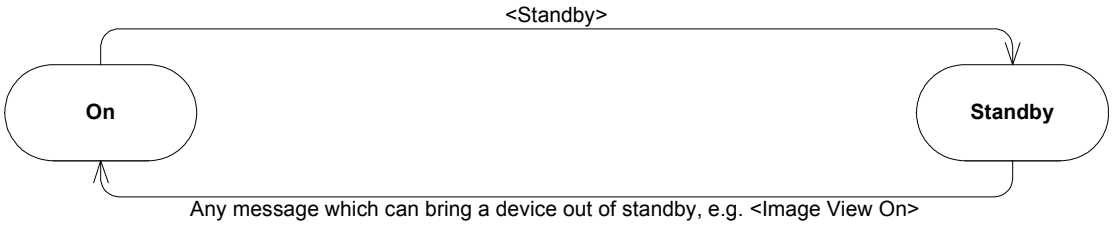
Playback Device: Deck Active, Deck Inactive

Menu Providing Device: Device Menu Active, Device Menu Inactive

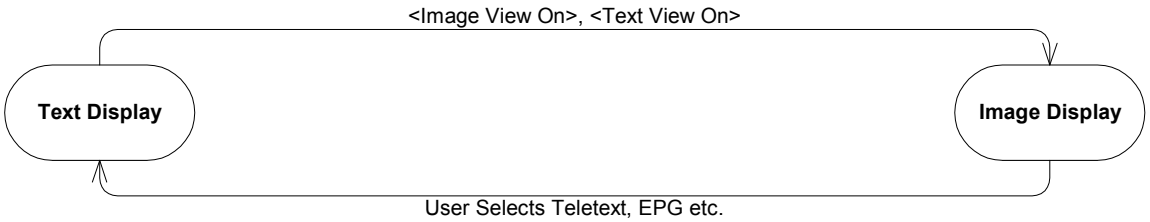
CEC 14.1.2 State Changes

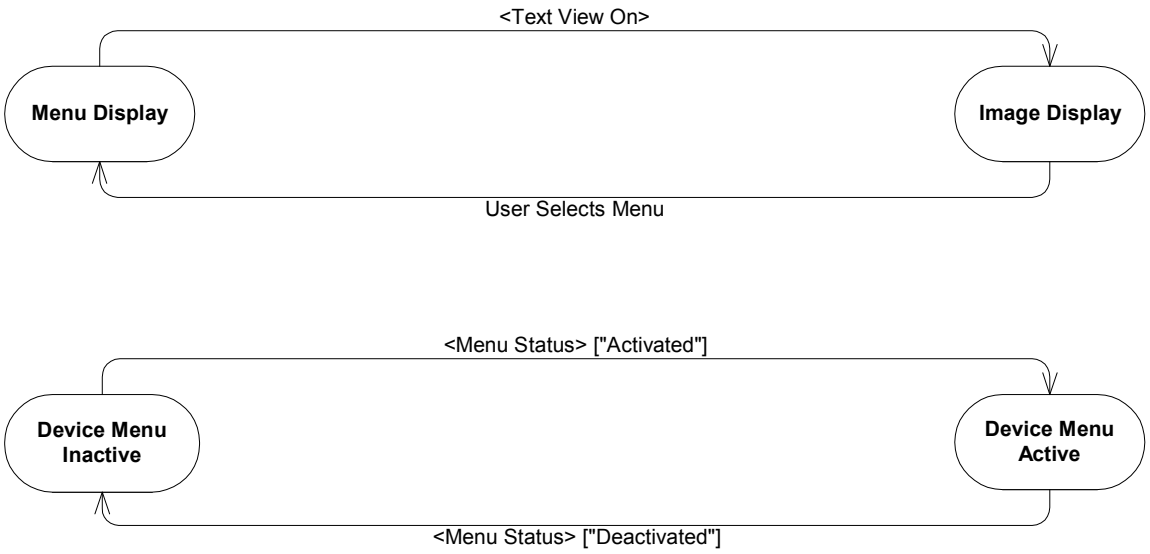
The following diagrams show the state transitions that are caused as a direct result of a device receiving a CEC message. Transitions between states that are not caused as a result of CEC messages are generally not shown, except where no CEC message can cause that transition.

CEC 14.1.3 All Devices

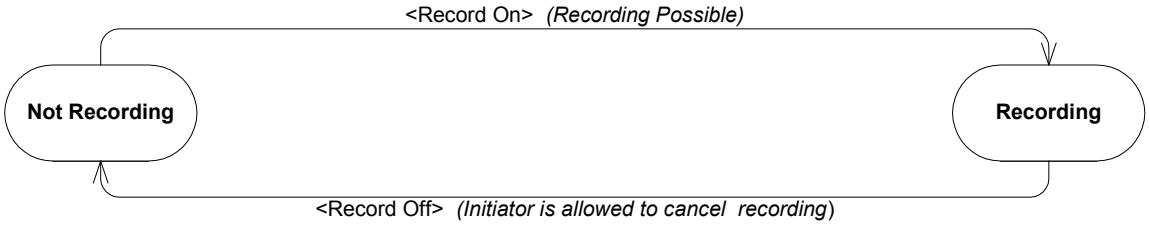


CEC 14.1.4 TV

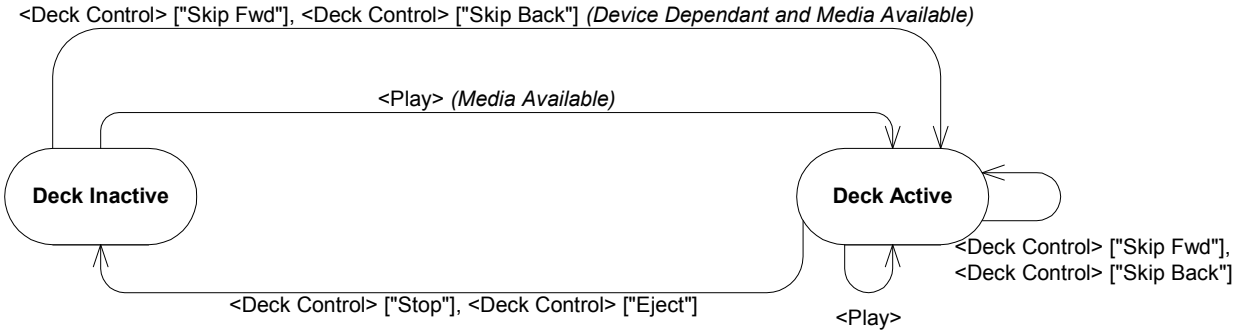




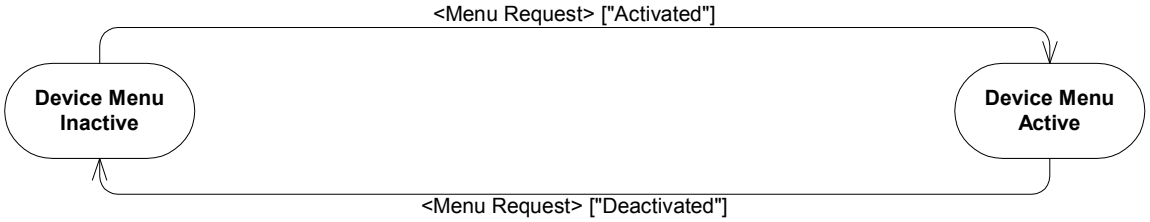
CEC 14.1.5 Recording Device



CEC 14.1.6 Playback Device



CEC 14.1.7 Menu Providing Device



CEC 15 Message Descriptions

The section defines the individual messages used in CEC. It describes them and defines their parameters and expected responses. As CEC has no session layer, this section and the operands section (CEC 17) effectively define the complete messaging system. Tables CEC Table 6 to CEC Table 18 show which messages are mandatory. If a manufacturer implements any of the optional messages, then they shall be implemented as described in CEC 13.

The following list describes each heading within the message tables CEC Table 6 to CEC Table 18.

- **Opcode** – The name used to identify the message.
- **Value** – The unique identifier for the message.
- **Description** – A brief description of the message.
- **Parameters** – The set of parameters used by the message, refer to CEC Table 22 for individual descriptions.
- **Parameter Description** – A brief description of the parameters that the message uses.
- **Response** – Describes how a device should respond on receipt of the message.
- **Directly Addressed** – Indicates if the message may be directly addressed.
- **Broadcast** – Indicates if the message may be broadcast.
- **Mandatory** – Indicates if it is mandatory for a device to react and respond on receipt of the message. Note that where a message is indicated as being mandatory for ‘All’ devices, this excludes devices which act only as a CEC Switch.

Within the table some cells are intentionally left blank; this indicates that there is no associated requirements for the Opcode described.

CEC Table 6 Message Descriptions for the One Touch Play Feature

Opcode	value	Description	Parameters	Parameter description	Response	Directly addressed	Broadcast	Mandatory for Initiator	Mandatory for Follower
<Active Source> [1]	0x82	Used by a new source to indicate that it has started to transmit a stream OR used in response to a <Request Active Source>	[Physical Address]	The physical address of the device.	A current active source should take appropriate action. TV should switch to the appropriate input. Any CEC switches between source and root should switch to the appropriate input and come out of standby if necessary.		•	All sources	TV, CEC Switches
<Image View On>	0x04	Sent by a source device to the TV whenever it enters the active state (alternatively it may send <Text View On>).	None		Turn on (if not on). If in 'Text Display' state then the TV enters 'Image Display' state. Note: Does not change TV menu or PIP status.	•		All sources shall implement at least one of <Image View On> or <Text View On>	TV
<Text View On>	0x0D	As <Image View On>, but also removes any text, menus and PIP windows from the TV's display.	None		As <Image View On>, but removes PIPs and menus from the screen. The TV enters 'Image Display' state regardless of its previous state.	•		All sources shall implement at least one of <Image View On> or <Text View On>	TV

[1] This message is also used in the Routing Control Feature

DELL/HP/LENOVO EXHIBIT 1015
Page 569

CEC Table 7 Message Descriptions for the Routing Control Feature

Opcode	value	Description	Parameters	Parameter description	Response	Directly addressed	Broadcast	Mandatory for Initiator	Mandatory for Follower
<Active Source> [2]	0x82	Used by a new source to indicate that it has started to transmit a stream OR used in response to a <Request Active Source>	[Physical Address]	The physical address of the device.	A current active source should take appropriate action. TV should switch to the appropriate input. Any CEC switches between source and root should switch to the appropriate input.		•	All sources	TV, CEC Switches
<Request Active Source>	0x85	Used by a new device to discover the status of the system.	None		<Active Source> from the present active source.		•		All except CEC Switches
<Routing Change>	0x80	Sent by a CEC Switch when it is manually switched to inform all other devices on the network that the active route below the switch has changed.	[Original Address] [New Address]	The previous child_address that the switch was switched to and the new child_address it has been moved to.	If a CEC Switch is at the new address, it sends a <Routing Information> message to indicate its current active route.		•	CEC Switches	CEC Switches
<Routing Information>	0x81	Sent by a CEC Switch to indicate the active route below the switch.	[Physical Address]	The current active route to the sink in the CEC Switch.	If a CEC Switch is at the specified address it should relay a <Routing Information> message to indicate its current active path.		•	CEC Switches	CEC Switches
<Set Stream Path>	0x86	Used by the TV to request a streaming path from the specified physical address.	[Physical Address]	The physical address of the source device.	Any CEC switches between should switch inputs according to the path defined in [Physical Address]. The device at the new address should stream its output and broadcast an <Active Source> message.		•		CEC Switches

[2] This message is also used in the One Touch Play Feature

DELL/HP/LENOVO EXHIBIT 1015
Page 570

CEC Table 8 Message Descriptions for the Standby Feature

Opcode	value	Description	Parameters	Parameter description	Response	Directly addressed	Broadcast	Mandatory for Initiator	Mandatory for Follower
<Standby>	0x36	Switches one or all devices into standby mode. Can be used as a broadcast message or be addressed to a specific device. It is recommended that a directly addressed message is only sent if the initiator originally brought the device out of standby. See section CEC 13.3 for important notes on the use of this message	None		Switch the device into standby. [3] Ignore the message if already in standby.	•	•		All

[3] Can be ignored if actively engaged in a recording or providing a source stream for a recording.

DELL/HP/LENOVO EXHIBIT 1015
Page 571

CEC Table 9 Message Descriptions for the One Touch Record Feature

Opcode	value	Description	Parameters	Parameter description	Response	Directly addressed	Broadcast	Mandatory for Initiator	Mandatory for Follower
<Record Off>	0x0B	Requests a device to stop a recording.	None		Exit 'Recording' state and stop recording if the initiator is the same as the initiator of the <Record On> message which started this recording.	•			Recording Device
<Record On>	0x09	Attempt to record the specified source.	[Record Source]	Source to record, either digital service or own source (ie currently selected source).	Enter 'Recording' state and start recording if possible. Send the initiator <Record Status>.	•			Recording Device, Own Source only
<Record Status>	0x0A	Used by a recording device to inform the initiator of the message <Record On> about its status.	[Record Status Info]	The recording status of the device.		•		Recording Device	Device Initiating a recording
<Record TV Screen>	0x0F	Request by the recording device to record the presently displayed source.	None		Initiate a recording using the <Record On> message, or send a <Feature Abort> ["Cannot provide source"] if the presently displayed source is not recordable.	•			

DELL/HP/LENOVO EXHIBIT 1015
Page 572

CEC Table 10 Message Descriptions for the System Information Feature

Opcode	value	Description	Parameters	Parameter description	Response	Directly addressed	Broadcast	Mandatory for Initiator	Mandatory for Follower
<Give Physical Address>	0x83	A request to a device to return its physical address.	None		<Report Physical Address>	•			All
<Get Menu Language>	0x91	Sent by a device capable of character generation (for OSD and Menus) to a TV in order to discover the currently selected Menu language. Also used by a TV during installation to discover the currently set menu language.	None		The addressed device responds with a <Set Menu Language> message	•			TV with OSD / Menu generation capabilities
<Polling Message>	-	Used by any device for device discovery – similar to ping in other protocols.	None		Shall set a low level ACK.	•		All except for TV and CEC Switches	
<Report Physical Address>	0x84	Used to inform all other devices of the mapping between physical and logical address of the initiator.	[Physical Address] [Device Type]	The device's physical address within the cluster.			•	All	TV
<Set Menu Language>	0x32	Used by a TV or anther device to indicate the menu language.	[Language]	The user's menu language choice.	Set the menu language as specified, if possible.		•	TV	All, except TV and CEC Switches

DELL/HP/LENOVO EXHIBIT 1015
Page 573

CEC Table 11 Message Descriptions for the Deck Control Feature

Opcode	value	Description	Parameters	Parameter description	Response	Directly addressed	Broadcast	Mandatory for Initiator	Mandatory for Follower
<Deck Control>	0x42	Used to control a device's media functions.	[Deck Control Mode]	The deck control requested.	Perform the specified actions, or return a <Feature Abort> message. It is device dependent whether or not a Skip Fwd/Skip Back (Wind/Rewind) command is legal when in the 'Deck Inactive' state. If the device is in standby and receives an eject command, it should power on and eject its media.	•			<Deck Control> ["Stop"] for a playback device
<Deck Status>	0x1B	Used to provide a deck's status to the initiator of the <Give Deck Status> message.	[Deck Info]	Information on the device's current status.		•			
<Give Deck Status>	0x1A	Used to request the status of a device, regardless of whether or not it is the current active source.	[Status Request]	Allows the initiator to request the status once or on all future state changes. Or to cancel a previous <Give Deck Status> ["On"] request.	<Deck Status>	•			
<Play>	0x41	Used to control the playback behaviour of a source device.	[Play Mode]	Play mode required.	Perform the specified actions, or return a <Feature Abort> message. If media is available the device enters 'Deck Active' state. If the device is in standby, has media available and the parameter is ["Play Forward"] it should power on.	•			<Play> ["Forward"] for a playback device

DELL/HP/LENOVO EXHIBIT 1015
Page 574

CEC Table 12 Message Descriptions for the Tuner Control Feature

Opcode	value	Description	Parameters	Parameter description	Response	Directly addressed	Broadcast	Mandatory for Initiator	Mandatory for Follower
<Give Tuner Device Status>	0x08	Used to request the status of a tuner device.	[Status Request]	Allows the initiator to request the status once or on all future state changes, or to cancel a previous <Give Tuner Device Status> [“On”] message.	Respond with a <Tuner Device Status> message, or stop reporting changes on receipt of the [“Off”] message.	•			
<Select Digital Service>	0x93	Directly selects a Digital TV, Radio or Data Broadcast Service	[Digital Service Identification]	Defines Digital TV system and necessary data to specify a service	Change to the selected digital service and stream its output on the HDMI connection. If the tuner device is not capable of selecting this service, respond with a <Feature Abort>	•			
<Tuner Device Status>	0x07	Use by a tuner device to provide its status to the initiator of the <Give Tuner Device Status> message.	[Tuner Device Info]	Information on the tuner devices current status.		•			
<Tuner Step Decrement>	0x06	Used to tune to next lowest service in a tuner’s service list. Can be used for PIP.	None		Follower tunes to next lowest service in its service list.	•		TV without Tuner and with Program Inc./ Dec. facility	Tuner Device
<Tuner Step Increment>	0x05	Used to tune to next highest service in a tuner’s service list. Can be used for PIP.	None		Follower tunes to next highest service in its service list.	•		TV without Tuner and with Program Inc./ Dec. facility	Tuner Device

DELL/HP/LENOVO EXHIBIT 1015
Page 575

CEC Table 13 Message Descriptions for the Vendor Specific Commands Feature

Opcode	value	Description	Parameters	Parameter description	Response	Directly addressed	Broadcast	Mandatory for Initiator	Mandatory for Follower
<Device Vendor ID>	0x87	Reports the vendor ID of this device.	[Vendor ID]	The vendor ID of the device.	Any other interested device may store the vendor ID of the device.		•	As needed for Devices supporting Vendor Specific Command	As needed for Devices supporting Vendor Specific Command
<Give Device Vendor ID>	0x8C	Requests the Vendor ID from a device.	None		<Device Vendor ID>	•		As needed for Devices supporting Vendor Specific Command	As needed for Devices supporting Vendor Specific Command
<Vendor Command>	0x89	Allows vendor specific commands to be sent between two devices.	[Vendor Specific Data]	Vendor specific command or data. The maximum length of the [Vendor Specific Data] shallnot exceed 14 data blocks to avoid saturating the bus.	Vendor Specific	•			
<Vendor Remote Button Down>	0x8A	Indicates that a remote control button has been depressed.	[Vendor Specific RC Code]	The vendor specific Remote Control Code for the key pressed.	Vendor Specific	•	•		
<Vendor Remote Button Up>	0x8B	Indicates that a remote control button (the last button pressed indicated by the Vendor Remote Button Down message) has been released.	None		Vendor Specific	•	•		

DELL/HP/LENOVO EXHIBIT 1015
Page 576

CEC Table 14 Message Descriptions for the OSD Status Display Feature

Opcode	value	Description	Parameters	Parameter description	Response	Directly addressed	Broadcast	Mandatory for Initiator	Mandatory for Follower
<Set OSD String>	0x64	Used to send a text message to output on a TV.	[Display Control] [OSD String]	Display timing. Text to be displayed.	TV displays the message.	•			

CEC Table 15 Message Descriptions for the Device OSD Transfer Feature

Opcode	value	Description	Parameters	Parameter description	Response	Directly addressed	Broadcast	Mandatory for Initiator	Mandatory for Follower
<Give OSD Name>	0x46	Used to request the preferred OSD name of a device for use in menus associated with that device.	None		<Set OSD Name>	•			
<Set OSD Name>	0x47	Used to set the preferred OSD name of a device for use in menus associated with that device.	[OSD Name]	The preferred name of the device.	Store the name and use it in any menus associated with that device.	•			

DELL/HP/LENOVO EXHIBIT 1015
Page 577

CEC Table 16 Message Descriptions for the Device Menu Control Feature

Opcode	value	Description	Parameters	Parameter description	Response	Directly addressed	Broadcast	Mandatory for Initiator	Mandatory for Follower
<Menu Request>	0x8D	A request from the TV for a device to show/remove a menu or to query if a device is currently showing a menu.	[Menu Request Type]	Indicates if the menu request is to activate or deactivate the devices menu, or to simply query the devices menu status.	May enter or exit the 'Device Menu Active' state if the parameter was "Activate" or "Deactivate" Send <Menu Status> to indicate the current status of the devices menu.	•			
<Menu Status>	0x8E	Used to indicate to the TV that the device is showing/has removed a menu and requests the remote control keys to be passed though.	[Menu State]	Indicates if the device is in the 'Device Menu Active' state or 'Device Menu Inactive' state.	If Menu State indicates activated, TV enters 'Device Menu Active' state and forwards those Remote control commands, shown in CEC Table 23, to the initiator. If deactivated, TV enters 'Device Menu Inactive' state and stops forwarding remote control commands.	•			
<User Control Pressed> [4]	0x44	Used to indicate that the user pressed a remote control button or switched from one remote control button to another.	[UI Command]	UI command issued by user.	Update display or perform an action, as required.	•			
<User Control Released> [5]	0x45	Indicates that user released a remote control button (the last one indicated by the <User Control Pressed> message)	None		Update display or perform an action, as required.	•			

[4] This message is also used in the RC Passthrough Feature
[5] This message is also used in the RC Passthrough Feature

DELL/HP/LENOVO EXHIBIT 1015
Page 578

CEC Table 17 Message Descriptions for the Remote Control Passthrough Feature

Opcode	value	Description	Parameters	Parameter description	Response	Directly addressed	Broadcast	Mandatory for Initiator	Mandatory for Follower
<User Control Pressed> [6]	0x44	Used to indicate that the user pressed a remote control button or switched from one remote control button to another.	[UI Command]	UI command issued by user.	Update display or perform an action, as required.	•			
<User Control Released> [7]	0x45	Indicates that user released a remote control button (the last one indicated by the <User Control Pressed> message)	None		Update display or perform an action, as required.	•			

CEC Table 18 Message Descriptions for the Power Status Feature

Opcode	value	Description	Parameters	Parameter description	Response	Directly addressed	Broadcast	Mandatory for Initiator	Mandatory for Follower
<Give Device Power Status>	0x8F	Used to determine the current power status of a target device	None		<Report Power Status>	•			
<Report Power Status>	0x90	Used to inform a requesting device of the current power status	[Power Status]	The current power status		•			

[6] This message is also used in the Device Menu Control Feature
[7] This message is also used in the Device Menu Control Feature

DELL/HP/LENOVO EXHIBIT 1015
Page 579

CEC Table 19 Message Descriptions for General Protocol messages

Opcode	value	Description	Parameters	Parameter description	Response	Directly addressed	Broadcast	Mandatory for Initiator	Mandatory for Follower
<Feature Abort>	0x00	Used as a response to indicate that the device does not support the requested message type, or that it cannot execute it at the present time.	[Feature Opcode] [Abort Reason]	The Opcode of the aborted message. The reason provides an indication as to whether the follower does not support the message, or does support the message but cannot respond at the present time.	Assume that request is not supported or has not been actioned.	•		Generate if a message is not supported	All
<Abort> Message	0xFF	This message is reserved for testing purposes.	None		A device shall never support this message, and shall always respond with a <Feature Abort> message containing any valid value for [Abort Reason]. CEC switches shall not respond to this message.	•			All, except for CEC switches

DELL/HP/LENOVO EXHIBIT 1015
Page 580

CEC 16 Message Dependencies

This section describes the dependencies between each message. This section is divided into two tables, which describe the following:

- CEC Table 20 describes the message dependencies when a device is capable of receiving a particular message (i.e. it does not <Feature Abort> indicating an ["Unrecognized opcode"] in response to the message).
- CEC Table 21 describes the message dependencies when a device is capable of sending a particular message.
- Each table describes the additional messages that the device shall be capable of receiving and sending if a particular message is supported.

CEC Table 20 Message dependencies when receiving a message

If device does not <Feature Abort> the following message:	**It shall not <Feature Abort> the message(s):**	**It shall be able to send the message(s):**
<Deck Control>	<Play>	<Deck Status>
<Deck Status>	-	<Play>, <Deck Control>
<Device Vendor ID> [8]		<Give Device Vendor ID>
<Get Menu Language>	-	<Set Menu Language>
<Give Deck Status>	<Play>, <Deck Control>	<Deck Status>
<Give Device Power Status	-	<Report Power Status>
<Give Device Vendor ID>	-	<Device Vendor ID>
<Give OSD Name>	-	<Set OSD Name>

[8] This message is broadcast and there is therefore no <Feature Abort>, but devices accepting this message shall also be able to send the associated message

DELL/HP/LENOVO EXHIBIT 1015
Page 581

If device does not <Feature Abort> the following message:	It shall not <Feature Abort> the message(s):	It shall be able to send the message(s):
<Give Physical Address>	-	<Report Physical Address>
<Give Tuner Device Status>	-	<Tuner Device Status>
<Image View On>	<Active Source> [9]	-
<Menu Request>	<User Control Pressed>, <User Control Released>	<Menu Status>
<Menu Status>	-	<Menu Request>, <User Control Pressed>, <User Control Released>
<Play>	<Deck Control>	<Deck Status>
<Record Off>	<Record On>	<Record Status>
<Record On>	<Record Off>	<Record Status>
<Record Status>	-	<Record On>, <Record Off>
<Record TV Screen>	<Record Status>	<Record On>, <Record Off>
<Request Active Source>[10]	-	<Active Source>
<Routing Change>[11]	<Routing Information>	-
<Routing Information>[12]	<Routing Change>	-
<Set Digital Service>	-	-
<Set Menu Language>	-	-
<Set OSD Name>	-	<Give OSD Name>
<Set OSD String>	-	-
<Set Stream Path>[13]	-	<Active Source> (not CEC Switches)

[9] This message is broadcast and there is therefore no <Feature Abort>, but devices shall also accept the associated message.
[10] This message is broadcast and there is therefore no <Feature Abort>, but devices accepting this message shall also be able to send the associated message.
[11] This message is broadcast and there is therefore no <Feature Abort>, but devices shall also accept the associated message.
[12] This message is broadcast and there is therefore no <Feature Abort>, but devices shall also accept the associated message.

DELL/HP/LENOVO EXHIBIT 1015
Page 582

If device does not <Feature Abort> the following message:	It shall not <Feature Abort> the message(s):	It shall be able to send the message(s):
<Text View On>[14]	<Active Source>	-
<Tuner Device Status>	-	<Give Tuner Device Status>
<Tuner Step Decrement>	<Tuner Step Increment>	-
<Tuner Step Increment>	<Tuner Step Decrement>	-
<User Control Pressed>	<User Control Released>	-
<User Control Released>	<User Control Pressed>	-
<Vendor Command>[15]	<Device Vendor ID>	<Give Device Vendor ID>
<Vendor Remote Button Down> [16]	<Vendor Remote Button Up>, <Device Vendor ID>	<Give Device Vendor ID>
<Vendor Remote Button Up> [17]	<Vendor Remote Button Down>, <Device Vendor ID>	<Give Device Vendor ID>

[13] This message is broadcast and there is therefore no <Feature Abort>, but devices accepting this message shall also be able to send the associated message.
[14] This message is broadcast and there is therefore no <Feature Abort>, but devices shall also accept the associated message
[15] This message is broadcast and there is therefore no <Feature Abort>, but devices shall also accept the associated message and also be able to send the associated message.
[16] This message can be broadcast and there may not be a <Feature Abort>, but devices shall also accept the associated messages and also be able to send the associated message.
[17] This message can be broadcast and there may not be a <Feature Abort>, but devices shall also accept the associated messages and also be able to send the associated message.

DELL/HP/LENOVO EXHIBIT 1015
Page 583

CEC Table 21 Message dependencies when sending a message

If device ever sends the following message:	**It shall be able to send the message(s):**	**It shall not <Feature Abort> the message(s):**
<Deck Control>	<Play>	-
<Deck Status>	-	<Give Deck Status>, <Play>, <Deck Control>
<Device Vendor ID>	-	<Give Device Vendor ID>
<Get Menu Language>	-	<Set Menu Language>[18]
<Give Deck Status>	<Play>, <Deck Control>	<Deck Status>
<Give Device Vendor ID>	-	<Device Vendor ID> [19]
<Give OSD Name>	-	<Set OSD Name>
<Give Physical Address>	-	<Report Physical Address> [20]
<Give Tuner Device Status>	-	<Tuner Device Status>
<Image View On>	<Active Source>	-
<Menu Request>	<User Control Pressed>, <User Control Released>	<Menu Status>
<Menu Status>	-	<Menu Request>, <User Control Pressed>, <User Control Released>
<Play>	<Deck Control>	-
<Record Off>	<Record On>	<Record Status>
<Record On>	<Record Off>	<Record Status>
<Record Status>	-	<Record On>, <Record Off>

[18] This message is broadcast and there is therefore no <Feature Abort>, but devices sending the associated message shall also be able to send this message.
[19] This message is broadcast and there is therefore no <Feature Abort>, but devices sending the associated message shall also be able to send this message.
[20] This message is broadcast and there is therefore no <Feature Abort>, but devices sending the associated message shall also be able to send this message.

DELL/HP/LENOVO EXHIBIT 1015
Page 584

If device ever sends the following message:	It shall be able to send the message(s):	It shall not <Feature Abort> the message(s):
<Record TV Screen>	<Record Status>	<Record On>, <Record Off>
<Report Power Status>	-	<Give Device Power Status>
<Request Active Source>	-	<Active Source>[21]
<Routing Change>	<Routing Information>	-
<Routing Information>	<Routing Change>	-
<Set Menu Language>	-	<Get Menu Language>
<Set OSD Name>	-	<Give OSD Name>
<Set Stream Path>	-	<Active Source> (not CEC Switches)
<Text View On>	<Active Source>	-
<Tuner Device Status>	-	<Give Tuner Device Status>
<Tuner Step Decrement>	<Tuner Step Increment>	-
<Tuner Step Increment>	<Tuner Step Decrement>	-
<User Control Pressed>	<User Control Released>	-
<User Control Released>	<User Control Pressed>	-
<Vendor Command>	<Device Vendor ID>	<Give Device Vendor ID>
<Vendor Remote Button Down>	<Vendor Remote Button Up>, <Device Vendor ID>	<Give Device Vendor ID>
<Vendor Remote Button Up>	<Vendor Remote Button Down>, <Device Vendor ID>	<Give Device Vendor ID>

[21] This message is broadcast and there is therefore no <Feature Abort>, but devices sending the associated message shall also be able to send this message.

DELL/HP/LENOVO EXHIBIT 1015
Page 585

CEC 17 Operand Descriptions

In the following table, Operand Descriptions are ordered alphabetically. Sub-operands, which only occur in a single parent operand, are grouped with their parent and are shown indented.

CEC Table 22 Operand Descriptions.

Name	Range Description		Length	Purpose
[Abort Reason]	"Unrecognized opcode"	0	1 byte	Reason for a <Feature Abort> response.
	"Not in correct mode to respond"	1		
	"Cannot provide source"	2		
	"Invalid operand"	3		
	"Refused"	4		
[ASCII digit]	0x30≤n≤0x39		1 byte	Subset of [ASCII] representing a printable digit character.
[ASCII]	0x20≤n≤0x7F		1 byte	Represents a printable character.
[Boolean]	"False"	0	1 bit	Flag
	"True"	1		
[Deck Control Mode]	"Skip Forward / Wind Forward"	1	1 byte	Used in <Deck Control>. Note: The "Skip Forward / Wind Forward" and "Skip Backward / Wind Backward" values are used for example in a DVD as next chapter and previous chapter and in a VCR as wind and rewind.
	"Skip Backward / Rewind"	2		
	"Stop"	3		
	"Eject"	4		

DELL/HP/LENOVO EXHIBIT 1015
Page 586

Name	Range Description		Length	Purpose
[Deck Info]	“Play”	0x11	1 byte	Indicates the current status of a tape or disk deck.
	“Record”	0x12		
	“Play Reverse”	0x13		
	“Still”	0x14		
	“Slow”	0x15		
	“Slow Reverse”	0x16		
	“Search Forward”	0x17		
	“Search Reverse”	0x18		
	“No Media”	0x19		
	“Stop”	0x1A		
	“Wind”	0x1B		
	“Rewind”	0x1C		
	“Index Search Forward”	0x1D		
	“Index Search Reverse”	0x1E		
	”Other Status”	0x1F		

DELL/HP/LENOVO EXHIBIT 1015
Page 587

Name			Range Description		Length	Purpose
[Device Type]			"TV"	0	1 byte	Allows additional devices, above the number allowed in the logical addressing mechanism, to indicate their device type.
			"Recording Device"	1		
			"STB"	3		
			"DVD"	4		
			"Audio System"	5		
[Digital Service Identification]			[Digital Broadcast System] [Service Identification]		7 bytes	Indicates Digital Broadcast System and the parameters to identify a specific service. Length of data depends on [Digital Broadcast System].
	[Digital Broadcast System]		"ARIB"	0x00	1 byte	Indicates the Digital Broadcast System of required service.
			"ATSC"	0x01		
			"DVB"	0x02		
	[Service Identification]		[ARIB data]		6 bytes	Specifies an ARIB digital service
			[ATSC data]		6 bytes	Specifies an ATSC digital service
			[DVB data]		6 bytes	Specifies a DVB digital service
	[ARIB data]		"Transport_stream_ID"		2 bytes	The transport_stream_ID of the transport stream carrying the required service
			"Service_ID"		2 bytes	The service_ID of the required service
			"Original_Network_ID"		2 bytes	The original_network_ID of the network carrying the transport stream for the required service
	[ATSC data]		"Transport_stream_ID"		2 bytes	The transport_stream_ID of the transport stream carrying the required service
			"Content_ID"		2 bytes	The content_ID of the required service
			"Reserved (0x0000)"		2 bytes	Reserved

DELL/HP/LENOVO EXHIBIT 1015
Page 588

Name		Range Description			Length	Purpose
	[DVB data]	“Transport_stream_ID”			2 bytes	The transport_stream_ID of the transport stream carrying the required service
		”Service_ID”			2 bytes	The service_ID of the required service
		”Original_Network_ID”			2 bytes	The original_network_ID of the network carrying the transport stream for the required service
[Display Control]		bit 5 – bit 0 = 0			1 byte	To indicate the display mode for an on screen display message.
			bit 7	bit 6		
		“Display for default time”	0	0		
		“Display until cleared”	0	1		
		“Clear previous message”	1	0		
		Reserved for future use	1	1		
[Feature Opcode]		0x00≤n≤0xFF (n is defined in CEC Table 6)			1 byte	Defines command to be performed.
[Language]		3 {[ASCII]} as defined in ISO/FDIS 639-2 [ref 1n]			3 bytes	Specify the language with which to interact with the user.
[Menu Request Type]		“Activate”		0	1 byte	Specifies whether to activate or deactivate a devices menu or simply query its current menu status.
		“Deactivate”		1		
		“Query”		2		
[Menu State]		“Activated”		0	1 byte	Specifies the state of a device menu
		“Deactivated”		1		
[New Address]		[Physical Address]			2 bytes	The physical address of the new device selected by a CEC Switch.
[Original Address]		[Physical Address]			2 bytes	The physical address of the device de-selected by a CEC Switch.

DELL/HP/LENOVO EXHIBIT 1015
Page 589

Name		Range Description		Length	Purpose
[OSD Name]		N {[ASCII]}, 1≤N ≤ 8		1 - 8 bytes	The devices name - To be used in On Screen Display references to it.
[OSD String]		N {[ASCII]}, 1≤N ≤ 13		1 - 13 bytes	A string to be displayed on the display.
[Physical Address]		4{[Port ID]}		2 bytes	Defines the path between the TV and a device – thus giving it a physical address within the cluster.
	[Port ID]	0x0≤n≤0xF		4 bits	Defines one 'hop' within the physical address relating to the physical connection of the device.
[Play Mode]		"Play Forward"	0x24	1 byte	The mode in which to play media. Note: If a device does not support a particular play mode it should select the closest match.
		"Play Reverse"	0x20		
		"Play Still"	0x25		
		"Scan Forward Min Speed"	0x05		
		"Scan Forward Medium Speed"	0x06		
		"Scan Forward Max Speed"	0x07		
		"Scan Reverse Min Speed"	0x09		
		"Scan Reverse Medium Speed"	0x0A		
		"Scan Reverse Max Speed"	0x0B		
		"Slow Forward Min Speed"	0x15		
		"Slow Forward Medium Speed"	0x16		
		"Slow Forward Max Speed"	0x17		
		"Slow Reverse Min Speed"	0x19		
		"Slow Reverse Medium Speed"	0x1A		
		"Slow Reverse Max Speed"	0x1B		

DELL/HP/LENOVO EXHIBIT 1015
Page 590

Name		Range Description		Length	Purpose
[Power Status]		"On"	0x00	1 byte	Used to indicate the current power status of a device.
		"Standby"	0x01		
		"In transition Standby to On"	0x02		
		"In transition On to Standby"	0x03		
[Record Source]		[Record Source Type] | {[Record Source Type] [Digital Service Identification]}		-	To define the source for a recording.
	[Record Source Type]	"Own source"	1	1 byte	Allows the record source to be specified for a recording.
		"Digital Service"	2		
[Record Status Info]		"Recording own source"	0x01	1 byte	Indicates the status of a recording.
		"Recording Digital Service Service"	0x02		
		"Reserved"	0x03		
		"Reserved"	0x04		
		"No recording – unable to record Digital Service "	0x05		
		"No recording - no media"	0x10		
		"No recording - playing"	0x11		
		"No recording - already recording"	0x12		
		"No recording - media protected"	0x13		
		"No recording - no source"	0x14		
		"No recording - media problem"	0x15		
		"No recording - other reason"	0x1F		

DELL/HP/LENOVO EXHIBIT 1015
Page 591

Name		Range Description		Length	Purpose
[Reserved Bit]		0		1 bit	Used as padding bit for future extensions.
[Status Request]		"On"	1	1 byte	Contains the status request mode which can be report once or on all future state changes or reporting off.
		"Off"	2		
		"Once"	3		
[Tuner Device Info]		[Tuner Display Info][Digital Service Identification]]		8 bytes	Indicates information about the tuner. Indicates the digital service that the tuner is set to, regardless of whether or not it is currently displaying the tuner.
	[Tuner Display Info]	"Displaying tuner"	0	1 byte	Indicates if the device is currently displaying its tuner or not (it may for example be displaying an external source or media).
		"Not displaying tuner"	1		
[UI Command]		0x00≤n≤0xFF (n is defined in CEC Table 23)		1 byte	Indicates the remote control button pressed.
[Vendor ID]		0x000000≤n≤0xFFFFFF (n is the 24-bit unique company ID [ref. 3i] obtained from the IEEE Registration Authority Committee (RAC)).		3 bytes	Identifier for a specific Vendor.

Notes: Items are transmitted in the order shown in the description.
All bit descriptions are done most significant bit first (i.e. first bit described is sent first)

DELL/HP/LENOVO EXHIBIT 1015
Page 592

CEC Table 23 User Control Codes

Operation id	User Operation
0x00	**Select**
0x01	**Up**
0x02	**Down**
0x03	**Left**
0x04	**Right**
0x05	**Right-Up**
0x06	**Right-Down**
0x07	**Left-Up**
0x08	**Left-Down**
0x09	Root Menu
0x0A	Setup Menu
0x0B	Contents Menu
0x0C	Favorite Menu
0x0D	**Exit**
0x0E - 0x1F	Reserved
0x20 - 0x29	**Numbers 0-9**
0x2A	Dot
0x2B	Enter
0x2C	Clear
0x2D - 0x2F	Reserved
0x30	Channel Up
0x31	Channel Down

Operation id	User Operation
0x32	Previous Channel
0x33	Sound Select
0x34	Input Select
0x35	Display Information
0x36	Help
0x37	Page Up
0x38	Page Down
0x39 - 0x3F	Reserved
0x40	Power
0x41	Volume Up
0x42	Volume Down
0x43	Mute
0x44	Play
0x45	Stop
0x46	Pause
0x47	Record
0x48	Rewind
0x49	Fast forward
0x4A	Eject
0x4B	Forward
0x4C	Backward
0x4D - 0x4F	Reserved

Operation id	User Operation
0x50	Angle
0x51	Sub picture
0x52 - 0x5F	Reserved
0x60	Play Function
0x61	Pause-Play Function
0x62	Record Function
0x63	Pause-Record Function
0x64	Stop Function
0x65	Mute Function
0x66	Restore Volume Function
0x67	Tune Function
0x68	Select Disk Function
0x69	Select A/V Input Function
0x6A	Select Audio Input Function
0x6B - 0x70	Reserved
0x71	**F1 (Blue)**
0x72	**F2 (Red)**
0x73	**F3 (Green)**
0x74	**F4 (Yellow)**
0x75	**F5**
0x76 - 0x7F	Reserved

Note: The elements identified in **bold** are the only ones which are forwarded as part of the device menu control feature.

DELL/HP/LENOVO EXHIBIT 1015
Page 593