



IEEE Standard for a High-Performance Serial Bus— Amendment 3

IEEE Computer Society

Sponsored by the
Microprocessor and Microcomputer Standards Committee

1394cTM

IEEE
3 Park Avenue
New York, NY 10016-5997, USA
8 June 2007

IEEE Std 1394cTM-2006
(Amendment to
IEEE Std 1394TM-1995)

IEEE Standard for a High- Performance Serial Bus— Amendment 3

Prepared by the

Microprocessor and Microcomputer Standards Committee
of the
IEEE Computer Society

Approved 6 December 2006
IEEE-SA Standards Board

Abstract: Supplemental information for a high-speed serial bus that integrates well with most IEEE standard 32-bit and 64-bit parallel buses is specified. It is intended to extend the usefulness of a low-cost interconnect between external peripherals. This standard follows the IEEE Std 1212-2001 command and status register (CSR) architecture.

Keywords: bus, computers, high-speed serial bus, interconnect

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2007 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 8 June 2007. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by the Institute of Electrical and Electronics Engineers, Incorporated.

Print: ISBN 0-7381-5237-4 SH95578
PDF: ISBN 0-7381-5238-2 SS95578

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

Use of an IEEE Standard is wholly voluntary. The IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEEE Standard document.

The IEEE does not warrant or represent the accuracy or content of the material contained herein, 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 material contained herein is free from patent infringement. IEEE Standards documents are supplied “**AS IS**.”

The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

In publishing and making this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon the advice of a competent professional in determining the exercise of reasonable care in any given circumstances.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position, explanation, or interpretation of the IEEE.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854
USA

Authorization to photocopy portions of any individual standard for internal or personal use is granted by the Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Introduction

This introduction is not part of IEEE Std 1394c-2006, IEEE Standard for a High-Performance Serial Bus—Amendment 3.

Work on this amendment was begun in January 2003, and the project was approved by the IEEE Standards Board in September 2003. The motivation for the work arose from the interest in the unshielded twisted pair (UTP) capabilities of the recently approved IEEE Std 1394b-2002, a desire to use higher speeds than S100, and the observation that the IEEE 802.3 gigabit Ethernet capabilities provided an excellent speed match to the encoded S800 data rate of IEEE Std 1394b and thus, in principle, could carry IEEE 1394b encoded data at this rate over 100 m of UTP cabling.

Networking issues were also considered; however, it was decided to focus the scope of work on extending IEEE Std 1394b by exploiting the gigabit Ethernet capabilities.

After careful study, the IEEE 1394 S800T Working Group decided to adopt an alternative to the 8B10B encoding scheme in IEEE Std 1394b, as the error properties of the gigabit Ethernet five-level pulse amplitude modulation (PAM5) encoding are significantly different from the properties of the nonreturn to zero (NRZ) transmission encoding used in IEEE Std 1394b, and a suitable encoding was devised. The detailed work included specifying the necessary first-in, first-out (FIFO) behaviors, updating low-power state machines, and extending the C code.

A major concern of the working group was to ensure good interoperability and user experience when connecting IEEE 802.3-based equipment to IEEE 1394c-based equipment. In particular, the specification supports the concept of allowing a connection on a device using an 8-pin modular connector (see 12.4.4) to operate either as an IEEE 802.3 connection or as a IEEE 1394c connection (IEEE Std 1394b provides no support for such “dual mode” connections). An extensive period of liaison took place with the relevant IEEE 802.3 committee to ensure a method of combining IEEE 1394c “negotiation” with IEEE 802.3 Auto-Negotiation that satisfies the requirements of IEEE 1394c without compromising IEEE 802.3 interoperability.

The technical work of the working group was concluded in July 2004.

Notice to users

Errata

Errata, if any, for this and all other standards can be accessed at the following URL: <http://standards.ieee.org/reading/ieee/updates/errata/index.html>. Users are encouraged to check this URL for errata periodically.

Interpretations

Current interpretations can be accessed at the following URL: <http://standards.ieee.org/reading/ieee/interp/index.html>.

Patents

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or

validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents or patent applications for which a license may be required to implement an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention. A patent holder or patent applicant has filed a statement of assurance that it will grant licenses under these rights without compensation or under reasonable rates and nondiscriminatory, reasonable terms and conditions to applicants desiring to obtain such licenses. The IEEE makes no representation as to the reasonableness of rates, terms, and conditions of the license agreements offered by patent holders or patent applicants. Further information may be obtained from the IEEE Standards Department.

Participants

The following is a list of participants in the IEEE 1394 S800T Working Group.

Michael D. Johas Teener, *Chair*

Richard Baker	Walter K. Hurwitz	Shiva Patibanda
Leslie A. Baxter	David V. James	Bill Russell
Patrick Beauperin	Peter G. Johansson	Mike Shinkarovsky
Kevin Brown	Prashant Kanhere	David Stone
John Creigh	Nick Kanzaki	David W. Thompson
Dwayne Escola	Sam Khoo	Richard Thousand
John Gildred	Jinhee Kim	Tom Truman
Carmen Gonzalez	Pascal Lagrange	Colin Whitby-Strevens
Don Harwood	Yoonsun Lee	Doug Williams
Burke Henahan		Johann Zipperer

The following members of the individual balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Eric W. Anderson	Fernando Genkuong	Steven G. Methley
Carlo Arpino	James P. Gilb	George J. Miao
Chris B. Bagge	Randall C. Groves	William J. Mitchell
Hugh Barrass	Pradeep Gupta	Michael S. Newman
Max Bassler	C. G. Guy	Vikram Punj
Leslie A. Baxter	Siamack Haghighi	Robert A. Robinson
Matthew K. Burnburg	Walter K. Hurwitz	Fernando Lucas Rodriguez
William A. Byrd	David Instone	Randall M. Safier
James T. Carlo	David V. James	Bartien Sayogo
Juan C. Carreon	Peter G. Johansson	Michael Scholles
Danila Chernetsov	Michael D. Johas Teener	James M. Skidmore
Kai Moon Chow	Piotr Karocki	Thomas E. Starai
Keith Chow	Stuart J. Kerry	David W. Thompson
Tommy P. Cooper	Yongbum Kim	Lars E. Thon
James R. Davis	Mark J. Knight	Carrie J. Tyer
Thomas J. Dineen	Joe Kring	Mark-Rene Uchida
James Dumser	David J. Law	Scott A. Valcourt
Sourav K. Dutta	Solomon Lee	Srinivasa R. Vemuru
Andrew C. Fieldsend	Daniel G. Levesque	Stephen C. Webb
Howard M. Frazier	Wei-Ting Lin	Colin Whitby-Strevens
John N. Fuller	G. L. Luri	David R. Wooten
Ignacio Marin Garcia	Ahmad Mahinfallah	Forrest D. Wright
Devon L. Gayle	Mark B. Maloney	Oren Yuen
Michael D. Geipel	Johnathon Meichtry	Janusz Zalewski

When the IEEE-SA Standards Board approved this standard on 6 December 2006, it had the following membership:

Steve M. Mills, *Chair*
Richard H. Hulett, *Vice Chair*
Don Wright, *Past Chair*
Judith Gorman, *Secretary*

Mark D. Bowman
Dennis B. Brophy
Joseph Bruder
Richard Cox
Bob Davis
Julian Forster*
Joanna N. Guenin
Mark S. Halpin
Raymond Hapeman

William B. Hopf
Lowell G. Johnson
Herman Koch
Joseph L. Koepfinger*
David J. Law
Daleep C. Mohla
Paul Nikolich

T. W. Olsen
Glenn Parsons
Ronald C. Petersen
Gary S. Robinson
Frank Stone
Malcolm V. Thaden
Richard L. Townsend
Joe D. Watson
Howard L. Wolfman

*Member Emeritus

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

Satish K. Aggarwal, *NRC Representative*
Richard DeBlasio, *DOE Representative*
Alan H. Cookson, *NIST Representative*

Jennie Steinhagen
IEEE Standards Program Manager, Document Development

Michael D. Kipness
IEEE Standards Program Manager, Technical Program Development

Contents

0.	Overview.....	1
0.1	Scope.....	1
0.2	Purpose.....	1
0.3	Document organization.....	2
1.	Overview.....	3
1.2	<u>Normative r</u> References.....	3
2.	Definitions and abbreviations.....	5
2.2	Technical.....	5
3.	Summary description.....	7
3.14	T-mode features.....	7
3.15	The relationship of T-mode to Beta mode.....	7
3.16	The relationship to IEEE Std 802.3-2005.....	7
3.17	S800 over UTP.....	7
3.18	Twin-mode ports.....	8
12.	~~CAT-5~~ UTP PMD ~~specification description~~	9
12.4	Media specification.....	9
14.	Connection management.....	11
14.1	Overview.....	11
14.2	Port characteristics.....	12
14.3	Functions, variables, and constants.....	13
14.4	Node-level port controller.....	17
14.5	Port connection manager state machine.....	17
14.6	Standby.....	22
14.7	Loop prevention.....	24
14.9	T-mode connectivity and operation.....	25
14.10	Simultaneous support for Beta mode and T-mode.....	25
14.11	Negotiation.....	25
15.	PHY register map.....	31
15.1	PHY register map for the cable environment.....	31
16.	Data routing, arbitration, and control.....	35
16.2	PHY services.....	35
16.3	PHY facilities.....	36
16.4	Cable PHY operation.....	36

19.	C code	55
19.1	Common declarations and functions.....	55
19.2	Connection management routines	71
19.3	Port state machine actions.....	96
19.4	Border arbitration actions and conditions	132
19.5	Border arbitration.....	169
20.	T-mode port specification	183
20.1	Overview	183
20.2	Port functions	184
20.3	T-mode port operation	191
21.	S800 UTP (T-mode) PMD electrical specification.....	199
21.1	T-mode PMD specification.....	200
21.2	T-mode PMD initialization	200
21.3	Gigabit media independent interface (GMII)	200
21.4	T-mode suspend and resume.....	201
21.5	UTP cable power.....	202

IEEE Standard for a High-Performance Serial Bus—Amendment 3

EDITORIAL NOTE—This standard amends the base IEEE 1394 document set, i.e., IEEE Std 1394™-1995 and its approved amendments IEEE Std 1394a™-2000 and IEEE Std 1394b™-2002. The editing instructions define how to merge the material in this amendment into the base IEEE 1394 document set to form the comprehensive standard as created by the addition of this amendment.

Editing instructions are shown in ***bold italic***. Four editing instructions are used: change, delete, insert, and replace. ***Change*** is used to make small corrections in existing text or tables. The editing instruction specifies the location of the change and describes what is being changed either by using ~~striethrough~~ (to remove old material) or underscore (to add new material). ***Delete*** removes existing material. ***Insert*** adds new material without disturbing the existing material. Insertions may require renumbering. If so, renumbering instructions are given in the editing instruction. ***Replace*** is used to make large changes in existing text, subclauses, tables, or figures by removing existing material and replacing it with new material. Editorial notes will not be carried over into future editions because the changes will be incorporated into the base standard.

0. Overview

0.1 Scope

This is a full-use standard whose scope is to amend the base IEEE 1394 document set (i.e., IEEE Std 1394-1995 as amended by IEEE Std 1394a-2000 and IEEE Std 1394b-2002) to specify alternate physical layer(s) (PHYs) that provide greater than S100 data rate over Category 5 (CAT-5) or better cable. This PHY is capable of negotiating with a peer device to select the appropriate next higher protocol layer.

0.2 Purpose

Technology advances have made gigabit signaling over unshielded twisted pair (UTP) cables a feasible and attractive extension to IEEE Std 1394. This extension will allow S800 (and possibly higher) video and audio networks to be built with a CAT-5 (or better) UTP cable plant instead of optical fiber. A negotiation method will be defined to allow peer devices to select the appropriate connection supported at both ends.

0.3 Document organization

This standard contains this overview, a list of definitions, a summary description, and clauses of technical specification. The new reader should read the definitions and summary description before reading the technical clauses.

1. Overview

Change the title of 1.2 as follows:

1.2 Normative rReferences

Insert the following reference into 1.2 in alphanumerical order:

IEEE Std 802.3™-2005, 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.¹

¹IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA (<http://standards.ieee.org/>).

2. Definitions and abbreviations

2.2 Technical

Insert the following into 2.2 in alphabetical order. Replace the “x” in 2.2.x with the appropriate numbers:

2.2.x T-mode: The mode in which the port is operating according to the specifications given in Clause 20 and Clause 21 of IEEE Std 1394.

2.2.x T-mode capable port: A port that is capable of operating in the mode specified in Clause 20 and Clause 21 of IEEE Std 1394.

2.2.x T-mode only port: A port that is not capable of Beta mode operation on unshielded twisted pair (UTP) at S100, but is capable of operating in the mode specified in Clause 20 and Clause 21 of IEEE Std 1394.

2.2.x twin-mode: The capability of a device to support the signalling defined in IEEE Std 1394 and the signalling defined in IEEE Std 802.3-2005² on a common media interface connector as specified in 12.4.4 of IEEE Std 1394.

2.2.x 802_mode: A port that is operating according to the specifications, including higher layer specifications, defined in IEEE Std 802.3-2005.

2.2.x unshielded twisted pair (UTP) cabling: Cabling that meets or exceeds the ISO/IEC 11801:2002 specifications for Class D balanced twisted pair cabling (which is equivalent to Category 5e cabling as specified in ANSI/TIA/EIA-568-B.2).

²Information on references can be found in 1.2.

3. Summary description

Insert the following new subclauses, 3.11 through 3.15, after 3.10:

3.11 T-mode features

This standard defines mechanisms that allow S800 IEEE 1394 connections over Category 5e or better unshielded twisted pair (UTP) cables and supports negotiation between peer devices to allow either a IEEE 1394 connection or an IEEE 802.3 connection.

This standard provides a mode of port operation known as T-mode. A port operating in T-mode transmits control, arbitration, and data as specified for Beta mode over UTP at S800 speeds.

This standard also updates some timing constants and algorithms in order to support larger diameter networks (up to 80 μ s round-trip time between any two nodes).

3.12 The relationship of T-mode to Beta mode

A T-mode port uses the arbitration tokens, control tokens, and packet formats defined for use in Beta mode unchanged. The encoding translates these into 10-bit symbols, using an encoding specific to T-mode. In particular, this encoding is unrelated to the 8B10B encoding used for Beta mode ports. BOSS arbitration as specified for Beta mode is used unchanged. A PHY may have any number of ports operating in data-strobe (DS) mode, Beta mode, or T-mode. The T-mode ports always operate at S800.

3.13 The relationship to IEEE Std 802.3-2005

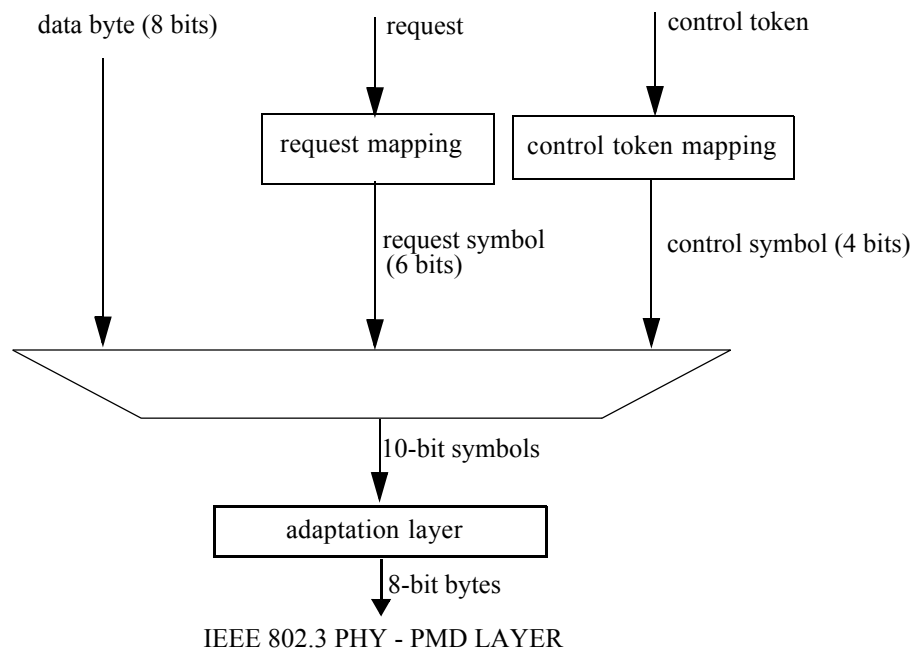
A T-mode port uses the PHY encoding/decoding defined in Clause 40 of IEEE Std 802.3-2005 to transmit Beta mode control, arbitration, and data over UTP. A T-mode port uses connection negotiation similar to IEEE 802.3 Auto-Negotiation, as defined in Clause 28, Annex 28A, Annex 28B, Annex 28C, Annex 28D, 40.5, and Annex 40C of IEEE Std 802.3-2005. On a new connection, T-mode port negotiation and IEEE 802.3 Auto-Negotiation run in parallel using distinct twisted pairs to determine the operating mode for the connection.

3.14 S800 over UTP

In the transmitting direction, a stream of data bytes, arbitration requests, and control signals is mapped to a stream of bytes that is in turn passed to the physical medium dependent (PMD) layer, as illustrated in Figure 3-35. The adaptation layer encodes data bytes, requests, and control tokens into 10-bit symbols. The resulting 10-bit symbol stream is then transmitted as a stream of IEEE 802.3 (Clause 40) 8-bit bytes, on a five-to-four basis, i.e., five 8-bit bytes are transmitted for every four 10-bit symbols.

In the receiving direction, a stream of data bytes is received from the PMD layer and converted back to the original sequence of data bytes, arbitration requests, and control signals. The encoding scheme allows a robust decode that can recover essential arbitration and control signaling in the presence of data byte errors.

The time taken to transmit a data byte, arbitration request, or control symbol is identical to a symbol time on a port operating according to Clause 13 at S800.

**Figure 3-35—T-mode coding functions**

3.15 Twin-mode ports

This standard allows for a port interfacing through a single media interface connector as specified in 12.4.4 to support both operation in T-mode and operation in 802_mode. Such a port is known as a twin-mode port. T-mode negotiation and IEEE 802.3 Auto-Negotiation take place in parallel, and the port selects the mode of operation appropriate to the peer port. 802_mode is selected if the peer device supports only the IEEE 802.3 specification; T-mode is selected if the peer port also supports T-mode.

Change the title of Clause 12 as follows:

12. ~~CAT-5~~ UTP PMD ~~specification~~ description

12.4 Media specification

12.4.4 Media interface connector

Replace Table 12-1 (to update IEEE 1394b S100 baseband two-pair signaling) with the following:

Table 12-1—Media interface connector pin assignments

Pin	Circuit
1	Twisted pair B (TPB)
2	TPB*
3	Twisted pair A (TPA)
4	—
5	—
6	TPA*
7	—
8	—

Replace 12.4.5 (to update IEEE 1394b S100 baseband two-pair signaling) with the following:

12.4.5 Autocrossover

The PMD layer shall support an autocrossover function, controlled by PMD_CONTROL_request.

On power reset and after a PMD_CONTROL.request(PMD_NO_CROSSOVER) from the connection management function, the PMD layer shall transmit on TPB and TPB* (pins 1 and 2, respectively, on the media interface connector) and shall receive on TPA and TPA* (pins 3 and 6, respectively, on the media interface connector).

After a PMD_CONTROL.request(PMD_CROSSOVER) from the connection management function, the PMD layer shall transmit on TPA and TPA* (pins 3 and 6, respectively, on the media interface connector) and shall receive on TPB and TPB* (pins 1 and 2, respectively, on the media interface connector).

The PMD layer shall report PH_AUTOCROSSOVER_SUPPORTED in response to a PMD_STATUS.request from the connection management function.

Note that Clause 14 was introduced in IEEE Std 1394b-2002.

14. Connection management

14.1 Overview

Replace Figure 14-1 with the following:

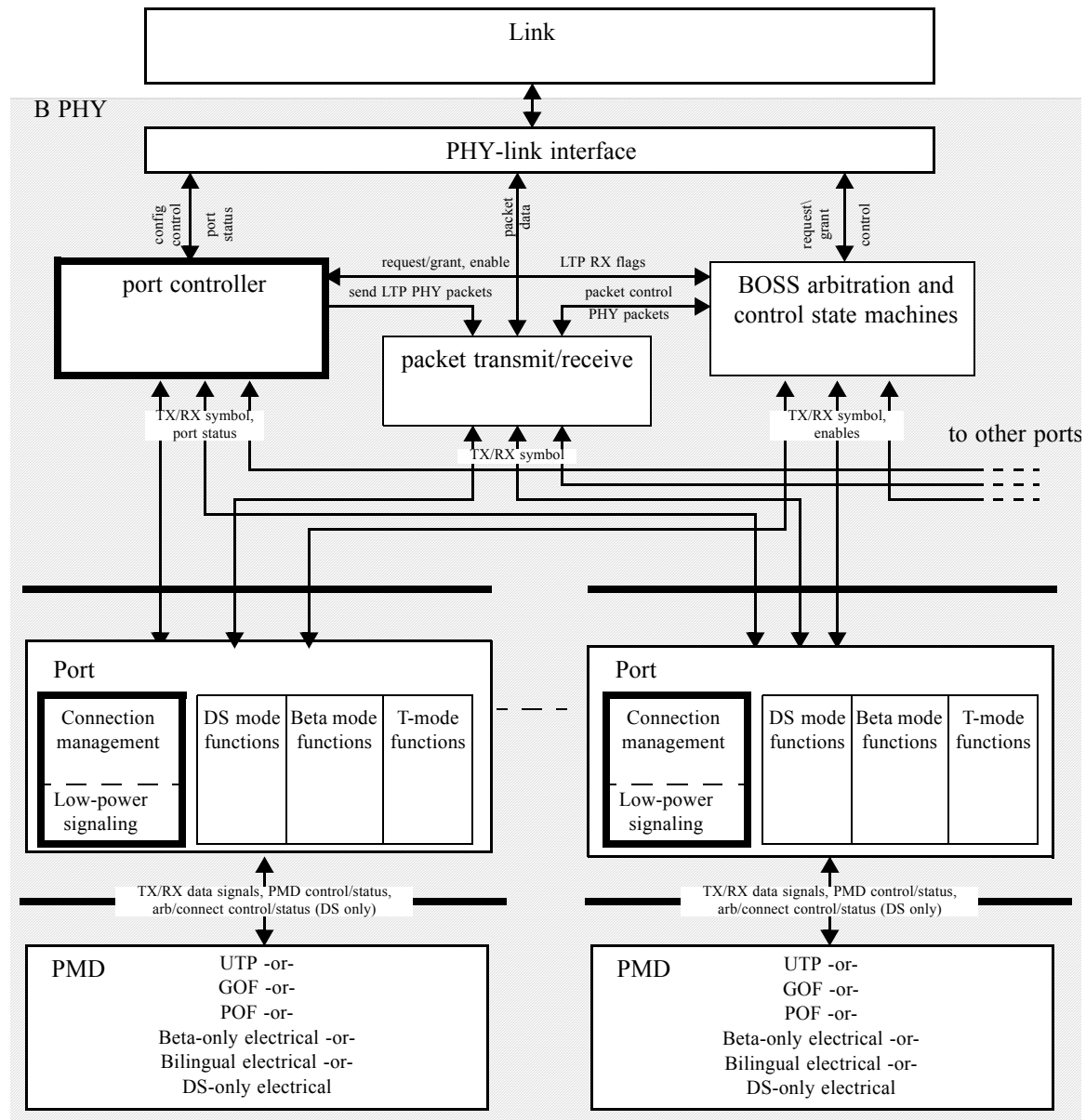


Figure 14-1—PHY master architecture (connection management in context)

Replace 14.2 through 14.6.2 (including Table 14-1, Table 14-2, Table 14-3, and Figure 14-2) with the following:

14.2 Port characteristics

The port is the device that controls the flow of data bits through the serial bus connections. The port's connection manager detects the presence or absence of its peer port, establishes reliable communications with the peer port in the most suitable mode of signaling, and manages low-power modes of operation, i.e., sleep modes.

14.2.1 Requirements

A T-mode capable port

- Shall detect and manage all connect and disconnect events.
- Shall be capable of operating in T-mode at S800.
- Shall be brought out to a media interface connector as specified in 12.4.4.
- Shall support suspend, resume, standby, restore, and low-power connection signalling.
- May be capable of operating in Beta mode at S100 (but at no other speed).
- When operating in T-mode, shall support all speeds of packets at or below S800 through the use of padding.

A Beta port

- Shall detect and manage all connect and disconnect events.
- Shall be capable of operating in Beta mode in a speed range compatible with the transmission medium (see Table 14-1); the lowest transmission rate in the port's speed range shall be no higher than the required speed, and the highest transmission rate shall be no higher than the maximum allowed speed.
- May be brought out to a Beta-only socket or a bilingual socket and shall not be brought out to a Legacy socket.
- When brought out to a bilingual socket, shall be capable of operating in DS mode at S100, S100 to S200, or S100 to S400.
- When brought out to a socket other than a media interface connector as specified in 12.4.4, shall report PMD_AUTOCROSSOVER_SUPPORTED as FALSE.
- Shall start up in Beta mode if Beta capable at a speed compatible with the Beta mode capability of the connected port and shall start up and run (i.e., set the operational speed) at the fastest such speed.
- In Beta mode, shall support all speeds of packets at or below its operational speed through the use of padding.
- Shall start up in DS mode if Beta mode is not possible, if both the port and the port to which it is connected are capable of DS operation, and if a dc connection exists between the ports.
- Shall support suspend, resume, standby, restore, and low-power connection signalling.

A DS port may be capable of operating at S100, S100 to S200, or S100 to S400. No requirement exists for the maximum speed of a port operating in DS mode to be related to the speed of the port or any other port in Beta mode.

Table 14-1—Media-dependent Beta mode speed requirements

Transmission medium	Required speed	Maximum allowed speed
Beta copper connector	S400 β	S3200 β
Media interface connector as specified in 12.4.4 and UTP cable	S100 β	S100 β
PN connector and POF fiber	S100 β	S200 β
Duplex LC connector and 50 μ m MMF	S400 β	S3200 β
NOTE—When a port is designed to be capable of supporting more than one medium, then the maximum speed may be set by writing to the <code>maxPortSpeed</code> register or other implementation-dependent mechanism.		

14.2.2 Properties

A Beta port may be connected directly to a suitable transceiver for long-haul connection (thus also providing dc isolation).

A Beta port may use dc isolation when operating in Beta mode.

14.3 Functions, variables, and constants

Connection management is specified as a port connection manager state machine for each port and a number of continuously running C code machines, some of which operate at node level. The C code functions and variables used in the port connection manager state machine are described in Table 14-2.

**Table 14-2—Variables and functions used in port connection manager state machine
(Sheet 1 of 2)**

Variable	Description
<code>active</code>	Indicates that the port is in the P2: Active state (set by state machine transitions to and from P2).
<code>betaMode</code>	Is set if the port has determined that it is operating in Beta mode; is unset otherwise (i.e., when operating in DS mode or when in the P0: Disconnected state). Is maintained as a read-only bit in the port register map.
<code>bportSyncOk</code>	Variable shared with the Beta port. Indicates that the port has byte and scrambler synchronization with its peer.
<code>connected</code>	Indicates that a connection is established with a peer port and (Beta mode only) operating speed negotiation is completed. Is maintained as a read-only bit in the port register map.
<code>connectionUnreliable</code>	PHY register bit. Is set to TRUE if the Beta mode speed negotiation fails or synchronization fails while establishing the operating mode and speed of a connection. Is reset by software after appropriate higher level action has been taken.
<code>disableRequest</code>	Requests transmission of <code>DISABLE_NOTIFY</code> at the end of the next packet.
<code>disabled</code>	PHY register bit. Is set to disable a port under software control, and indicates that the port is in the P6: Disabled state.

**Table 14-2—Variables and functions used in port connection manager state machine
(Sheet 2 of 2)**

Variable	Description
doStandby	Is true when the port has been commanded to be a standby initiator.
forceDisconnect	Is true to force disconnect and retest of a port after detecting a loop during reset.
inStandby	Indicates the port is in the P7: Standby Initiator, P8: Standby Target, P9: Standby, or P10: Restore state.
loopDisabled	Indicates the port is in the P12: Loop Disabled state.
loopToDetect	Is true during reset up to S1: Self-ID Grant or S2: Self-ID Receive state. (See Figure 16-9.)
receiveOk	In DS mode, indicates the reception of a debounced TpBias signal. In Beta mode, indicates the reception of a continuous electrically valid signal. Is set to FALSE during the time that only connection tones are detected in Beta mode. Is maintained as a read-only bit in the port register map (supersedes the <i>Bias</i> bit).
restore	Indicates that the port is restoring. Also causes a port in standby to restore.
resume	Indicates that the port is resuming. Also causes a suspended port to resume.
resumeFault	Is set when the peer port fails to complete resume.
rxDisabled	Indicates the reception of a DISABLE_NOTIFY token from the peer port.
rxStandb	Indicates the reception of a STANDBY token from the peer port.
rxSuspend	Indicates the reception of a SUSPEND token from the peer port.
signaled	Indicates the transmission of a DISABLE_NOTIFY, SUSPEND, or STANDBY token.
standbyFault	Is set when the peer port fails to complete standby.
suspendRequest	Requests the transmission of a SUSPEND token at the end of the next packet and/or indicates that the port is suspending.
suspendFault	Is set when the peer port fails to complete suspend.
tPort802Mode	Is true when Auto-Negotiation has selected IEEE 802.3 higher layers.
untested	Indicates the port is connected, but untested.
untestedState	Indicates the port is in the P11: Untested state.

The timing constants and related parameters used in the connection management descriptions are defined in Table 14-3.

Table 14-3—Connection management constants (Sheet 1 of 2)

Timing constant	Minimum	Maximum	Comment
BIAS_FILTER_TIME	41.6 μ s	52.1 μ s	Time on a DS-capable port to filter <code>biasDetect</code> upon the detection of <code>TpBias</code> before updating the PHY register <code>ReceiveOk</code> bit ($\sim 4096/BASE_RATE$).
COLLISION_LIMIT	7		Number of times a collision condition occurs to conclude that a loop exists.
CONNECT_TIMEOUT	330 ms	350 ms	Connection debounce time.
DISCONNECTED_TONE_INTERVAL	42.66 ms	48 ms	$TONE_DURATION * 4 * TONE_INTERVAL$ ($2^{**22}/BASE_RATE$).
MAX_OCCUPANCY_TIME	83.975 μ s	83.993 μ s	The duration of the maximum occupancy timer (the maximum time that the bus is held while performing loop-free build to allow a peer connection to issue a short bus reset) ($\sim 8256/BASE_RATE$).
NUM_OF_TRIES	4		Number of tries at toning before switching to trying <code>TpBias</code> .
PORT_ENABLE_TIME		1 ms	Time necessary for <code>TpBias</code> output to become valid when driven high. The minimum value is implementation-dependent and may incorporate other design timing requirements for port resumption.
RECEIVE_OK_HANDSHAKE	5.3 ms	16 ms	When used to detect a fault during a suspend or resume handshake on a port operating in DS mode, either the time a suspend initiator waits for the suspend target to remove bias (measured from the transmission of <code>TX_SUSPEND</code>) before a suspend fault exists or the time a resume initiator waits for the resume target to assert bias (measured from the generation of <code>TpBias</code> by the resume initiator) before a resume fault exists. Also the time a suspend target waits, measured from the time it drives <code>TpBias</code> low, before it suspends. Ports operating in Beta mode may also use this timing value instead of values based on Beta mode synchronization time.
RECEIVER_INIT_TIME		1 ms	Time from first receipt of signal for a receiver to be operating within the bit error ratio (BER) objective of 10^{-12} .
RESET_DETECT	80.0 ms	85.3 ms	Time for an active port to confirm a reset signal.

Table 14-3—Connection management constants (Sheet 2 of 2)

Timing constant	Minimum	Maximum	Comment
RESUME_CHECKS	65		$\text{DISCONNECTED_TONE_INTERVAL} / \text{TONE_DURATION} + 1$, the number of checks that are made between connection tone intervals.
SPEEDCODE_BIT_INTERVAL	1.998 ms	2.013 ms	$\text{TONE_DURATION} * 3$, the time between the end of one tone position and the start of the next.
SYNCHRONIZATION_LENGTH		16384	(Maximum number of transmitted bits for scrambler synchronization + byte synchronization rounded up to a power of two).
TEST_INTERVAL	41.66 μ s	41.67 μ s	Time to allow a test value to be returned as a loop test symbol (LTS) ($\text{LONG_BOSS_RESTART_TIME} / 2$).
TONE_DURATION	666 μ s	678 μ s	$2^{**16} / \text{BASE_RATE}$.
TONE_GAP	8		$\text{TONE_INTERVAL} / 2$, the number of tone units in the gap before the start bit.
TONE_INTERVAL	16		The number of tone units in a tone interval.
TMODE_DISCONNECT_TIMEOUT	1.6 s	1.8 s	Timeout for the detection of a disconnection after loss of synchronization causing entry to suspend on a T-mode port.
TPORT_MAX_LATENCY	0.96 ms	1.26 ms	Time to wait for a SUSPEND, DISABLE or STANDBY token to propagate to a peer T-mode port.
TPORT_MIN_SUSPEND_TIME	96 ms	126 ms	Time to wait during entry to suspend, standby, or loop disabled to ensure a minimum time before exiting these states.
TPORT_SUSPEND_HANDSHAKE	1.6 s	1.8 s	When used to detect a fault during a suspend or resume handshake on a port operating in T-mode, either the time a suspend initiator waits for the suspend target to remove normal signaling (measured from the transmission of TX_SUSPEND) before a suspend fault exists or the time a resume initiator waits for the resume target to resume signaling (measured from the command to resume signaling by the resume initiator) before a resume fault exists,
NOTE—The values of TEST_INTERVAL and COLLISION_LIMIT are chosen to ensure that the loop testing will complete much more than a test interval before the MAX_OCCUPANCY_TIME timer expires. This ensures that the final loop test packet (LTP) with ATTACH_IN_PROGRESS is seen by all controlling nodes on other buses before an attach is completed.			

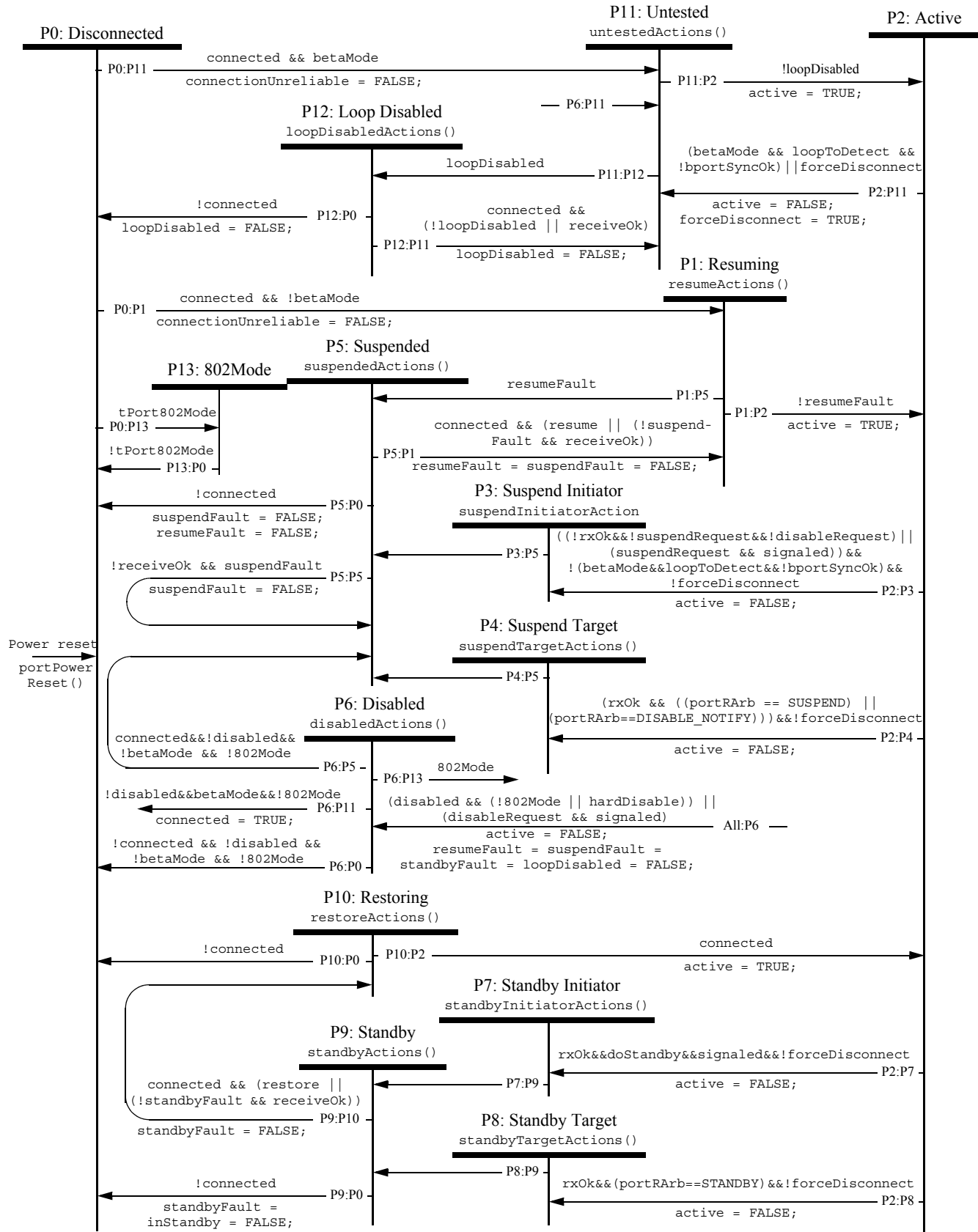
14.4 Node-level port controller

One instantiation of the node-level port controller shall occur in a PHY. Its behavior is specified as continuously running C code and maintains a record of whether the node is an isolated or border node. It also contains routines called by the individual port connection managers providing a global view of the connection information for the other ports. Finally, it provides bias debounce for each port using a shared timer. It shares (by read-only access) one element of each of the array variables `active[NPORT]`, `connected[NPORT]`, `disabled[NPORT]`, `suspend[NPORT]`, `resume[NPORT]`, `bias[NPORT]`, `proxy[NPORT]`, `restore[NPORT]`, and `betaMode[NPORT]` and the array constants `BETA_MODE_ONLY_PORT[NPORT]` and `T_MODE_PORT[NPORT]` with the respective port connection managers. Note that, in the C code relating to the port controller, these variables are subscripted; but in the C code relating to the individual port connection manager, the variables have the same name but are not subscripted.

14.5 Port connection manager state machine

The port connection manager state machine operates independently for each port. A port may be capable of operation only in DS mode (4.2.2, 4.2.3, 4.3.1, 4.3.2, and 4.3.3), only in Beta mode (Clause 13), only in T-mode (Clause 20), in both DS mode and Beta mode (i.e., a bilingual port), or in both Beta mode and T-mode. In all cases, while a port is in the active state, its arbitration, data transmission, reception, and repeat behaviors are specified by the state machines in Clause 16. When a PHY port is in any state other than active, it is permissible for it to lower its power consumption; the only functional component of a PHY that shall be active in all states is the physical connection detect circuitry or toning circuitry, as appropriate.

Figure 14-2 illustrates the port connection manager state machine, and following the figure are notes applicable to this state machine.



Power reset. A power reset of the PHY initializes each port as disconnected.

Transition All:P6. This transition applies to a port in any of the P states (other than P6) and takes priority over other available transitions. The local link may immediately disable a port in any state other than P13: 802Mode by setting the disabled bit to 1. A port in P13 shall remain in P13 until the `tPort802Mode` flag is cleared or both the `disabled` and `hardDisable` flags are set, and then the port shall take the transition into P6 if the disabled flag is set to 1. This transition may also be caused by a remote command packet, in which case the port, if active, shall transmit `DISABLE_NOTIFY` before the port is disabled.

State P0: Disconnected. The generation of `TpBias` is disabled (DS mode), or continuous transmission is stopped (Beta mode); and the outputs are in a high-impedance state. The PHY may place most of the port's circuitry in a low-power consumption state. A port shall transmit tones, and its signal detect and (if DS capable) connection detect mechanism shall be active even if other components of the PHY port are in a low-power state. If the port is a `T_MODE_CAPABLE` port and toning indicates that its peer PHY port is physically connected, it engages in Auto-Negotiation with the peer port to select either T-mode operation or 802_mode operation.

Transition P0:P11. When toning signals that a port's peer PHY port is physically connected, the PHY port transitions to the P11: Untested state.

Transition P0:P13. When the result of Auto-Negotiation is that the port is to operate in 802_mode, the port transitions to the P13: 802Mode state. It appears as a disconnected port for the purpose of IEEE 1394 operation.

State P1: Resuming. In Beta mode, the PHY commences transmission at the operating speed and attempts to synchronize its receiver. The PHY port tests both the connection status and the presence of `receiveOk` to determine whether normal operations may be resumed. If the port is connected, `receiveOk` is set, and no other ports are active, the PHY waits seven `RESET_DETECT` intervals before any state transitions. Otherwise, in the case of a border node with one or more active ports, the PHY waits three `RESET_DETECT` intervals before any state transitions. A detected bus reset overrides either of these waits; otherwise, when the wait elapses, the PHY initiates a bus reset.

Transition P1:P2. If the PHY port did not fault during the resume handshake, it waits for bus reset to start and then transitions to the P2: Active state.

Transition P1:P5. A resuming PHY port that faults during the resume handshake transitions to the P5: Suspended state.

State P2: Active. The PHY port is fully operational and capable of transmitting or receiving and of repeating arbitration signals or clocked data. While the port remains active, it shall comply with the cable arbitration states specified in Clause 16.

Transition P2:P3. Upon the loss of `receiveOk` or the receipt of a PHY remote command packet that sets the `suspend` variable to 1, the PHY port leaves the P2: Active state to start functioning as a suspend initiator. A loss of `receiveOk` is usually the result of a physical disconnection or the loss of power to the connected peer PHY port. If the transition is the result of a remote command packet, the PHY transmits a remote confirmation packet with the `ok` bit set to 1. In the meantime, the suspend initiator has signaled `SUSPEND` to its connected peer PHY.

Transition P2:P4. If an active port observes an `DISABLE_NOTIFY` or `SUSPEND` signal, it becomes a suspend target and leaves the P2: Active state.

Transition P2:P7. Upon the receipt of a PHY remote command packet that sets the `doStandby` variable to 1, the PHY port leaves the P2: Active state to start functioning as a standby initiator.

Transition P2:P8. If an active port observes a STANDBY signal, it becomes a standby target and leaves the P2: Active state.

State P3: Suspend Initiator. A suspend initiator waits for `receiveOk` to be zero. If `RECEIVE_OK_HANDSHAKE` elapses and the connected peer PHY has not either driven `TpBias` low (DS mode) or ceased sending a valid signal (Beta mode), then the suspend operation has failed and the fault bit is set to 1. In either case, the suspend initiator first drives `TpBias` low for `RECEIVE_OK_HANDSHAKE` time if in DS mode and then places all outputs in a high-impedance state. When in T-mode, the suspend initiator ceases signaling after `TPORT_MAX_LATENCY`, waits for up to `TPORT_SUSPEND_HANDSHAKE` for the peer to withdraw signaling (setting the fault bit if the peer signaling is not withdrawn within this time), and waits for `TPORT_MIN_SUSPEND_TIME` to ensure that the port is not resumed immediately after being placed in suspend state.

Transition P3:P5. Upon completion of the actions associated with this state, the PHY port unconditionally transitions to the P5: Suspended state.

State P4: Suspend Target. A suspend target sets the `suspend` variable for all the other active ports. This step in turn causes the other active ports to propagate the `SUSPEND` signal. In the meantime, the suspend target (if in DS mode) drives its `TpBias` outputs below 0.1 V or (if in Beta mode) ceases sending a valid signal in order to signal the suspend initiator that `SUSPEND` was detected. If in DS mode, the node waits a `RECEIVE_OK_HANDSHAKE` time to allow the connected peer PHY time to drive `TpBias` low, and then the suspend target disables the generation of `TpBias`. It then places the outputs in a high-impedance state. When in T-mode, the suspend target ceases signaling, waits for up to `TPORT_SUSPEND_HANDSHAKE` for the peer to withdraw signaling, and waits for `TPORT_MIN_SUSPEND_TIME` to ensure that the port is not resumed immediately after being placed in suspend state.

Transition P4:P5. Upon completion of the actions associated with the P4: Suspend Target state, the PHY port unconditionally transitions to the P5: Suspended state.

State P5: Suspended. The PHY may place most of the port's circuitry in a low-power consumption state. The port shall maintain connectivity status by (Beta mode) transmitting tones and using its signal detect circuitry or (DS mode) using its connection detect circuit even if other components of the PHY port are in a low-power state.

Transition P5:P0. A suspended PHY port that loses its physical connection to its peer PHY port transitions to the P0: Disconnected state.

Transition P5:P1. Either of two conditions cause a suspended PHY port to transition to the P1: Resuming state:

- A nonzero value for the port's `resume` variable.
- The detection of `receiveOk` if the port's `suspendFault` variable is zero.

A port's `resume` variable may be set indirectly as the result of the resumption of other PHY ports.

Transition P5:P5. If the port entered the P5: Suspended state in a faulted condition (i.e., `receiveOk` was still present), the fault is cleared if and when `receiveOk` is removed by the peer PHY.

State P6: Disabled. While the port is in the P6: Disabled state, the PHY may place most of the port's circuitry in a low-power consumption state. If the `hardDisable` flag is `FALSE` and the port was operating in DS mode, the connect detect circuit shall be active even if other components of the PHY port are in a low-power state. If the `hardDisable` flag is `FALSE` and the port was operating in Beta mode, then it shall maintain connectivity information by toning. If the `hardDisable` flag is `TRUE`, then the port shall not

engage in toning. If the port is T-mode capable and toning indicates that its peer PHY port is physically connected, it engages in Auto-Negotiation with the peer port to select either T-mode operation or 802_mode operation.

Transition P6:P0. If the disabled bit is zero and the PHY port is not physically connected to its peer PHY port, it transitions to the P0: Disconnected state.

Transition P6:P5. If the disabled bit is zero and the PHY port is connected and operating in DS mode, it transitions to the P5: Suspended state.

Transition P6:P11. If the disabled bit is zero and the PHY port is connected and operating in Beta mode, it transitions to the P11: Untested state.

Transition P6:P13. If the result of Auto-Negotiation indicates that the port is to operate in 802_mode, it transitions to the P13: 802Mode state, regardless of the setting of the disabled bit.

Transition P7:P9. Upon completion of the actions associated with the P7: Standby Initiator state, the PHY port unconditionally transitions to the P9: Standby state.

State P7: Standby Initiator. A standby initiator waits for `receiveOk` to be zero. If `RECEIVE_OK_HANDSHAKE` elapses and the connected peer PHY has not ceased sending a valid signal, the standby operation has faulted and the `standbyFault` bit is set to 1. As soon as either `receiveOk` goes to zero or `RECEIVE_OK_HANDSHAKE` elapses, the standby initiator places all outputs in a high-impedance state. When in T-mode, the standby initiator ceases signaling after `TPORT_MAX_LATENCY`, waits for up to `TPORT_SUSPEND_HANDSHAKE` for the peer to withdraw signaling (setting the fault bit if the peer signaling is not withdrawn within this time), and waits for `TPORT_MIN_SUSPEND_TIME` to ensure that the port is not restored immediately after being placed in suspend state.

State P8: Standby Target. A standby target sets the `proxy` flag to indicate that it will proxy for the peer node. It ceases sending a valid signal in order to signal the standby initiator that `STANDBY` was detected. Then it places the outputs in a high-impedance state. When in T-mode, the standby target ceases signaling, waits for up to `TPORT_SUSPEND_HANDSHAKE` for the peer to withdraw signaling, and waits for `TPORT_MIN_SUSPEND_TIME` to ensure that the port is not restored immediately after being placed in standby.

Transition P8:P9. Upon completion of the actions associated with the P8: Standby Target state, the PHY port unconditionally transitions to the P9: Standby state.

State P9: Standby. The PHY may place most of the port's circuitry in a low-power consumption state. The port shall maintain connectivity status by (Beta mode) transmitting tones and using its signal detect circuitry or (DS mode) using its connection detect circuit even if other components of the PHY port are in a low-power state.

Transition P9:P0. A PHY port in the P9: Standby state that loses its physical connection to its peer PHY port transitions to the P0: Disconnected state.

Transition P9:P10. Either of two conditions causes a PHY port in the P9: Standby state to transition to the P10: Restoring state:

- A nonzero value for the port's `restore` variable.
- The detection of `receiveOk` if the port's `standbyFault` variable is zero as a result of failure to complete the previous suspend handshake.

A port's `restore` variable may be set indirectly as the result of the resumption of other PHY ports.

State P10: Restoring. The PHY commences transmission at the operating speed and attempts to synchronize its receiver. The PHY port tests both the connection status and the presence of `receiveOk` to determine whether normal operations may be restored. If `proxy` is set, then the PHY arbitrates for the bus and sends a restore configuration packet on the port. When the peer PHY indicates that it is in phase (i.e., by receiving appropriate arbitration requests), the restore operation is complete. If `proxy` is not set, then the PHY (necessarily a leaf node) awaits reception of the configuration packet and sets its phase variables appropriately.

Transition P10:P2. If the PHY port did not fault during the restore handshake, it transitions to the P2: Active state.

Transition P10:P0. A restoring PHY port that faults during the restore handshake ceases transmission on the port, waits for an apparent disconnection to be detected, and then transitions to the P0: Disconnected state.

State P11: Untested. In the P11: Untested state, the port is a candidate for testing to ensure that a loop would not be formed by completing the connection. The port starts up, synchronizes with its peer, and exchanges LTSs with its peer. It also transmits an `ATTACH_REQUEST` when required to do so by the node-level loop-free build mechanism and reports any incoming `ATTACH_REQUEST`. It completes any attachment by waiting for or generating a reset as appropriate.

Transition P11:P2. On completion of an attachment, the port is marked as active.

Transition P11:P12. If the node-level loop-free build mechanism determines that completing the connection would result in a loop or if synchronization is lost during the loop-free build process (`untestedFault`), then the port transitions to the P12: Loop Disabled state.

State P12: Loop Disabled. The port is disabled, but continues to tone (Beta mode), to send LTP signals (T-mode), or to use its signal detect circuitry (DS mode) to maintain connectivity status. A port remains in this state until a bus reset is detected, a resume signal is received from the peer port, or a disconnection is detected.

Transition P12:P11. A port in the P12: Loop Disabled state transitions to the P11: Untested state when a bus reset is detected or a resume signal is received from the peer port.

Transition P12:P0. A port in the P12: Loop Disabled state transitions to the P0: Disconnected state when disconnection is detected.

State P13: 802Mode. A port in the P13: 802Mode state appears as a disconnected port for IEEE 1394 operations. The disabled bit has no effect when the port is in the P13: 802Mode state unless the `hardDisabled` bit is also set.

Transition P13:P0. A port in the P13: 802Mode state transitions to the P0: Disconnected state when `802_mode` operations cease.

14.6 Standby

Standby is a term used to describe a low-energy-consumption mode of operation for a port. Standby is a feature of Beta mode operation only. If a node has only one active port, then the connection on this port can be placed into standby; and, while the connection is in standby, the node does not participate in normal bus activity. Other nodes on the bus of which the node is a member are not aware of any status change of the node. A bus reset does not occur as part of entering or restoring from standby. The node is designated the *nephew*, and its peer node the *uncle*. The uncle's peer port is also placed in standby. Should a bus reset occur while the connection is in standby, the uncle node proxies a self-ID packet on behalf of the nephew node.

14.6.1 Nephew node characteristics

A port shall enter into standby and the local node become a nephew when the port receives a standby packet containing the node ID of the local node unless the local node is root, it has no active port, it has more than one connected port, it has a Legacy link, its `enableStandby` flag is zero, or the one active port is not operating in Beta mode. In any of these exception cases, the command shall be rejected. A node should not command another node to enter standby if the candidate node is performing bus management functions (e.g., bus manager, isochronous resource manager, cycle master).

A write to the `ibr` register on a nephew shall be ignored (the link is not able to initiate a bus reset).

A port in standby on a nephew node shall restore from standby either on receipt of any bus request from the local link or on detecting a restore tone. It shall transmit a `PH_Event.indication` of `PH_RESTORE_NO_RESET` to the link and shall establish synchronization with the uncle port. When a nephew port restores from standby, the time from the receipt of the link request or the restore tone until the port is ready to receive a restore packet shall not exceed 3 ms. The PHY shall assume (possibly incorrectly) a bus phase and, as soon as synchronization is achieved, shall transmit `NONE_*` requests of this phase on the port. A restoring port becomes active after receiving the following information from the first received restore packet:

- a) Node ID
- b) Gap count
- c) Gap count reset disable status
- d) Bus phase
- e) Reset notify (bus reset occurred while the port was in standby)

The PHY shall update its internal values and flags with this information, except that if Reset notify is not set, then the PHY shall not update its local copy of Node ID, but shall continue to use the previously determined value. The PHY shall not repeat this packet on the interface between the PHY and link or between the PHY that is integrated into a link (PIL) and fan-out PHY (FOP). As soon as the PHY has received the restore packet, it shall take control of the bus and transmit a S100 Beta mode null packet (`DATA_PREFIX` symbols). If the link or PIL interface is active, the PHY shall wait until it has completed the transfer of the status information to the attached link or PIL. It shall then terminate the null packet by performing the normal BOSS actions for end of subaction.

A multiport node with one connected port in standby shall restore that port followed by a bus reset if it detects a new connection on any of its other ports.

14.6.2 Uncle node characteristics

A PHY shall store sufficient information from each self-identification sequence for each port to allow it to proxy the self-ID packet for a nephew PHY attached to that port on a subsequent bus reset. When the node detects a standby symbol on a port, it assumes the role of uncle. It shall put the port into standby and shall proxy the self-ID packet of the nephew node on the occurrence of any bus reset that occurs while the port is in standby. A uncle node shall restore the nephew (i.e., assert a resume tone to the nephew) when it receives a Restore PHY command packet with the node ID of the uncle node and the port address of the port connected to the nephew node or when it detects a restore tone from the nephew.

NOTE—If errors occur in a self-identification sequence, then a candidate uncle node may not be able to store appropriate information to allow it to proxy. Software on any node that may invoke the standby mechanism should verify that each received self-identification sequence is valid and, if not valid, cause a bus reset.³

³Notes in text, tables, and figures are given for information only, and do not contain requirements needed to implement the standard.

When either detecting or generating a restore tone from or to a nephew node, the uncle node shall establish synchronization on the port connected to the nephew. As soon as synchronization is established, the uncle shall forward its favorite request to the nephew. As soon as synchronization is established, the uncle shall arbitrate for the bus and, when granted, shall generate a Restore PHY command packet, which it sends only to the nephew, and send a null packet to all active ports and the local link. The Restore PHY packet shall be terminated with DATA_END symbols to allow the nephew to assume the role of BOSS. It shall then mark the port as active. The uncle node shall service only one restoring nephew leaf at a time.

14.7 Loop prevention

Replace 14.7.8 with the following:

14.7.8 Loop test

When a node becomes the controlling node, it resets and starts its maximum occupancy timer and sends a first LTP. Each time it sends an LTP, it also resets and starts the test timer. The test timer times the test interval, which starts when an LTP is sent on the active bus and expires after TEST_INTERVAL. The duration of the test interval allows the LTP to propagate to the furthest extremes of the bus and for an LTS with the same loop test data (LTD) to be received on the test port. The test interval starts when an LTP is sent on the active bus.

During the test interval, every node with untested ports compares the LTD in the LTS received on the test port to the values in the holding register (HR). As long as the LTD in the received LTS is not equal to the HR contents and the mode of the received LTS is not ATTACH_IN_PROGRESS, the test continues for the duration of the test interval. If, however, at any time during the test interval, the LTD in the LTS is equal to the HR, then a collision condition exists. Upon detecting a collision, the controlling node shall select a new test_value number, send a new LTP, and reset and restart the test interval timer. The controlling node shall repeat this process up to COLLISION_LIMIT times while continuing to hold the bus unless or until further testing might result in the node holding the bus for more than MAX_OCCUPANCY_TIME. If the controlling node detects COLLISION_LIMIT collisions in a row, then it shall place the test port in the loop disabled state and release the bus. If at any time during the test the mode of the received LTS is ATTACH_IN_PROGRESS, then the test of the port is abandoned (i.e., the controlling node sends a null packet and releases the bus).

When a node has control of the bus in order to perform loop testing, has detected a possible collision, and is about to issue a new LTP, and if further testing might cause the node to hold the bus for more than MAX_OCCUPANCY TIME, then the node shall send a null packet, release the bus, and re-request the bus. When it regains control of the bus, and provided that the test of the port has not been abandoned, it shall issue a new LTP and continue testing the port, using the count of collisions from the previous set of tests. See 16.4.9.2 for a description of the implications for large diameter networks.

When the test interval expires without a collision being detected or without the mode of the received LTS being equal to ATTACH_IN_PROGRESS, the controlling node shall check to see whether it is still dominant over the node at the other end of the connection on the test port. If the HR in the controlling node is less than the HR of the LTS from the test port, then the other node is dominant and the test is abandoned (i.e., the controlling node sends a null packet and releases the bus). If the controlling node is dominant, then it shall attempt to complete the attach.

After 14.8.4, add the following new subclauses, 14.9 through 14.11.3.5.2:

14.9 T-mode connectivity and operation

When instructed to negotiate by the port state machine, the PMD layer shall commence negotiation as described in 14.11 and shall continue with negotiation as described in 14.11 until the negotiation either succeeds or fails. If a status confirmation of PMD_TPORT_OK and PMD_TPORT_FW is received from the PMD layer at the completion of negotiation, then the port shall set both the `betaMode` and `tMode` flags. The port state machine functions are then followed as in 14.5. When instructed to start normal operation, the port shall use the encoding specified in Clause 20. From the point of view of the higher levels, there is no difference between Beta mode and T-mode operation, apart from latency, jitter, and behavior when a port is placed in `hardDisable` and the behavior when negotiation results in `802_mode`.

14.10 Simultaneous support for Beta mode and T-mode

A port may be capable of supporting both Beta mode and T-mode. Such a port may be connected to a peer port that can operate either as a port that supports S100 Beta mode only, another port that also supports both modes, or a port that supports T-mode only. In the first case, the port shall operate in Beta mode at S100, and in the second two cases, the port shall operate in T-mode at S800.

When instructed to negotiate by the port state machine, the PMD layer shall commence negotiation as described in 14.11 and shall also enable its signal detect circuitry (it shall not commence toning at this stage). It shall cease negotiation upon detecting a toning signal compliant with 14.8 on pair 1 and 2 or pair 3 and 6, and it shall report a status of `SIGNAL_DETECT`. Otherwise, it shall continue with negotiation as described in 14.11 until the negotiation either succeeds or fails.

On receiving a status of `SIGNAL_DETECT`, the port state machine shall set the port maximum and minimum speeds to S100 and shall commence Beta connectivity toning and speed negotiation as specified in 14.8, followed by synchronization and normal Beta mode operation. The PMD layer shall support autocrossover. Autocrossover shall be attempted if necessary as described in 12.4.5.

14.11 Negotiation

When the PMD layer is instructed by the port state machine to negotiate (by means of the PMD_CONTROL_request of PMD_START_NEGOTIATE), the port shall commence to seek a peer connection and negotiate the mode of operation. During the time that the PMD layer is seeking a connection or negotiating, it shall provide a confirmation to a PMD_STATUS_request of `NEGOTIATION_ACTIVE`. At the completion of negotiation, the PMD layer shall provide a confirmation of PMD_TPORT_OK and PMD_TPORT_FW, or PMD_TPORT_802, or none of these (no connection or negotiation failed).

14.11.1 Overview

Devices using this standard are designed to coexist on the same UTP cables used by IEEE 802.3 10/100/1000Base-T Ethernet.

A negotiation protocol is used to ensure that the following situations resolve appropriately:

- a) T-mode only port connected to T-mode only port
- b) T-mode only port connected to T-mode + Ethernet twin-mode port
- c) T-mode + Ethernet twin-mode port connected to T-mode + Ethernet twin-mode port
- d) T-mode only port connected to Auto-Negotiation capable Ethernet only port
- e) T-mode only port connected to Legacy Ethernet only port

The negotiation scheme determines whether a common mode of operation exists and, if so, enables the best performing common mode. If no common mode of operation exists, the devices will not force bad packets into the network or otherwise interfere with a correctly functioning network.

The negotiation scheme chosen borrows heavily from IEEE 802.3 Auto-Negotiation and extends it to T-mode ports.

A T-mode port sends a simplified base page with a T-mode selector field. This is followed by Message Code 8 Next Pages. Then Message Code 5 Next Pages is sent containing the organizationally unique identifier (OUI) for this standard (00-19-0D). These pages are sent and received on twisted pairs C and D while optionally allowing Ethernet negotiation to proceed in parallel on twisted pairs A and B. After each side's capabilities are exchanged, the highest performance common technology is enabled automatically.

14.11.2 S100 Beta mode parallel negotiation

A device that complies with Clause 12 and Clause 13 and supports S100 Beta mode only on a port will transmit tones (per 14.8) on TPB, i.e., pins 1 and 2 on the media interface connector as specified in 12.4.4 for that port, which will be received by the peer device on TPA (pins 3 and 6). A device compliant with this standard may support S100 Beta mode on the same port as it supports T-mode and IEEE 802.3 operation. Reception of these tones by a PHY that supports S100 Beta mode will cause SIGNAL_DETECT to assert (per 14.8).

14.11.2.1 Clause 22 in IEEE Std 802.3-2005

The media independent interface (MII) register space as defined in 22.2.4 in IEEE Std 802.3-2005 is dedicated to IEEE 802.3 Auto-Negotiation. No bits are allocated for IEEE 1394 operation within the MII register space. An implementation may use an equivalent parallel set of registers to control T-mode negotiation as a method of implementing the negotiation services of PMD_START_NEGOTIATE (see 16.2.5.1) and PMD_STATUS_confirmations (see 16.2.5.2). An implementation may provide an implementation-dependent method of exposing direct access to the parallel set of registers for test purposes, e.g., by providing access via management data input/output (MDIO) at a different (nonzero) PHY address (PHYAD).

Access to the IEEE 802.3 management registers has no effect on T-mode port operation. In particular, PHY reset (setting bit 0.15 to a logic one) shall not disrupt T-mode port operation.

14.11.3 Differences between T-mode and IEEE 802.3 negotiation

With the exceptions described in 14.11.3.1 through 14.11.3.5, T-mode negotiation is Auto-Negotiation as defined in the following sections of IEEE Std 802.3-2005:

- Clause 28
- Annex 28A
- Annex 28B
- Annex 28C
- Annex 28D
- 40.5
- Annex 40C

14.11.3.1 Clause 28 in IEEE Std 802.3-2005**14.11.3.1.1 IEEE 802.3 (Clause 28) interface to media dependent interface (MDI)**

IEEE 802.3 operation relies on Auto-Negotiation using the standard 8-pin modular connector of IEC 60603-7 and its transmission and reception assignments as described in 14.5.1 in IEEE Std 802.3-2005. This requires transmission on pair A (contacts 1 and 2) and reception on pair B (contacts 3 and 6).

To eliminate conflict and speed cross-standard negotiation, a T-mode port uses the alternate two pairs (pairs C and D) for transmission and reception. Table 40-12 in IEEE Std 802.3-2005 shows the full contact to pair mapping for all four pairs. For a T-mode port, transmission shall be done on MDI pair C (BI_DC±), and reception shall be done on MDI pair D (BI_DD±). Support for crossover cabling correction shall be provided by allowing pairs C and D to be swapped.

14.11.3.1.2 IEEE 802.3 (28.3) Auto-Negotiation arbitration state machine(s)

Twin-mode ports supporting T-mode and Ethernet shall implement negotiation as described in this standard.

A T-mode only port shall implement negotiation as described in this standard.

This enhanced negotiation for twin-mode devices requires two concurrent negotiations: normal Ethernet Auto-Negotiation on copper cable pairs A and B and IEEE 1394 negotiation on pairs C and D.

Therefore, two instances of the Auto-Negotiation arbitration state machine specified by Figure 28-16 in IEEE Std 802.3-2005 are required to operate at the same time. Since there are two instantiations, it is also required to provide an interlock between them. The two Auto-Negotiation state machines shall operate as specified in IEEE Std 802.3-2005, except as specified in the remainder of this clause.

The results of the two parallel negotiations are evaluated as the state machines enter the respective FLP LINK GOOD CHECK states (see 28.3 of IEEE Std 802.3-2005), according to the modified priority resolution table (see 14.11.3.3). If T-mode is selected, then the Ethernet management registers shall be set to appear as if the (Ethernet) port is disconnected.

If the negotiation parameters are changed on one of the pairs, negotiation shall restart on both pairs.

As a matter of notation, each of the state machines accesses the variables of the other state machine by prepending `o_` to the variable name.

14.11.3.1.3 Subclause 28.3.1 in IEEE Std 802.3-2005

A variable with “`_ [S100Beta]`” appended to the end of the variable name indicates a variable or set of variables for which a IEEE 1394 S100 Beta mode physical medium attachment (PMA) is the signal source.

For parallel detection of a S100 Beta mode only device, the assertion of `SIGNAL_DETECT` shall assert the `link_status_ [S100Beta] = READY` primitive as defined in 28.2.6.1 in IEEE Std 802.3-2005, and the IEEE 802.3 Auto-Negotiation algorithm shall behave as specified in IEEE Std 802.3-2005 except as follows:

The single `link_ready` variable is a status variable indicating that `flp_receive_idle = true` and only one the of the following indications is being received:

```
link_status_ [NLP] = READY
link_status_ [TX] = READY
link_status_ [T4] = READY
```

link_status_[S100Beta] = READY

Values: false: Either zero or more than one of the above indications are true, or flp_receive_idle = false.
true: Exactly one of the above indications is true, and flp_receive_idle = true.

The mr_parallel_detection_fault is an error condition indicating that while performing parallel detection, either flp_receive_idle = false, or zero or more than one of the following indications were present when the autoneg_wait_timer expired. This signal is cleared on read of the Auto-Negotiation expansion register.

link_status_[NLP] = READY

link_status_[TX] = READY

link_status_[T4] = READY

link_status_[S100Beta] = READY

Values: false: Exactly one of the above indications was true when the autoneg_wait_timer expired, and flp_receive_idle = true.
true: Either zero or more than one of the above indications was true when the autoneg_wait_timer expired, or flp_receive_idle = false.

14.11.3.1.4 Figure 28-16 in IEEE Std 802.3-2005

In order to support parallel detection of S100 Beta mode, the following is added to the transition conditions from ABILITY DETECT to LINK STATUS CHECK:

+ link_status_[S100Beta] = READY

In order to provide an interlock between the two arbitration state machines, the following are added:

- a) A new state

```

SYNC WAIT
-----
waiting <= true

```

Waiting is true when in the SYNC WAIT state and cleared upon exit.

- b) A state transition from COMPLETE ACKNOWLEDGE to SYNC WAIT with the following conditions:

double_negotiation=true * {original COMPLETE ACKNOWLEDGE to FLP LINK GOOD CHECK conditions}

double_negotiation indicates whether two simultaneous Auto-Negotiations are occurring: one on pair A and B and the other on pair C and D. It takes a value of true when two negotiations are being performed and false when only one negotiation is being performed. This variable is set by the configuration of the device.

NOTE—This will put an upper bound on the number of extra Next Pages sent on one pair but not the other.

- c) An activity timer with an expiration period of 4 s and started in the ACKNOWLEDGE_DETECT state.

- d) A state transition from SYNC WAIT to LINK GOOD CHECK with the following conditions:

o_waiting + o_activity_timer_done * o_break_link_timer_done * o_autoneg_wait_timer_done

- e) The state transition from COMPLETE ACKNOWLEDGE to FLP LINK GOOD CHECK modified as follows:

double_negotiation=false * {original COMPLETE ACKNOWLEDGE to FLP LINK GOOD CHECK conditions}

- f) The ABILITY DETECT state with the following addition:

PMD_Status_Request=NEGOTIATION_ACTIVE==true

- g) The FLP LINK GOOD state with the following addition:

`PMD_Status_Request=NEGOTIATION_ACTIVE==false`

When negotiation is performed over pairs C and D, the arbitration state diagram states LINK STATUS CHECK and PARALLEL DETECTION FAULT are unused. It is not expected that Legacy technologies would transmit on pairs C or D.

14.11.3.2 Annex 28A in IEEE Std 802.3-2005

Annex 28A in IEEE Std 802.3-2005 shall be used as is except that the selector value of 00100₂ shall be used for T-mode negotiation.

14.11.3.3 Annex 28B in IEEE Std 802.3-2005

Annex 28B in IEEE Std 802.3-2005 shall be used as is except for the following changes:

- a) The Selector field $S[4:0] = 00100$.
- b) The Technology Ability field is reserved. It shall be transmitted as 0 and ignored on reception.
- c) The Extended Next Page bit is reserved. It shall be transmitted as 0 and ignored on reception.
- d) The Remote Fault bit in the T-mode base page shall be used as defined in 28.2.1.2.3 in IEEE Std 802.3-2005.
- e) The Ack bit in the T-mode base page shall be used as defined in 28.2.1.2.4 in IEEE Std 802.3-2005.
- f) The Next Page bit in the T-mode base page, defined in 28.2.1.2.4 in IEEE Std 802.3-2005, shall be always set to 1.
- g) T-mode takes priority above 1000Base-T in the priority resolution table.
- h) As part of the highest common denominator (HCD) selection process, `PMD_TPORT_FW` shall be set to true when T-mode is selected, and `PMD_TPORT_802` shall be set to true when an IEEE 802.3 technology (e.g., 100BASE-T, 1000BASE-T) has been selected.

14.11.3.4 Annex 28C in IEEE Std 802.3-2005

Annex 28C in IEEE Std 802.3-2005 shall be used as is except for the following changes:

- a) Message Codes 0–31 are as defined by IEEE Std 802.3-2005. All other codes are reserved.
- b) Message Code 5 is used to transmit OUIs. It shall be sent with the OUI for this standard (00-19-0D) as part of standard T-mode negotiation.

14.11.3.4.1 IEEE 802.3 (28C.6) Message Code 5 (OUI tag code)

For T-mode, the OUI tagged message shall consist of a single message code of 0000 0000 0101₂ followed by four user codes. The IEEE-assigned OUI value is 00-19-0D for this standard, and the manufacturer-selected extension identifier for T-mode is 0000000000₂ or 1000000000. The message code values generated from these two numbers is encoded into four message codes, as specified in Figure 14-6. For clarity, the position of the global broadcast *g* is illustrated.

NOTE—Figure 14-6 shows the order in which Next Pages are transmitted, with the first transmitted Next Page shown in the leftmost position. The bits within each individual Next Page, however, are transmitted starting with bit D0 and finishing with bit D15.

Within this illustration, the *a* bit indicates whether T-mode is supported, as specified in Table 14-5. The *r* bits are reserved: these shall be transmitted as 0 and ignored on reception.



Value	Name	Description
0	UNSUPPORTED	T-mode is not supported
1	SUPPORTED	T-mode is supported

Note that Clause 15 was introduced in IEEE Std 1394b-2002.

15. PHY register map

15.1 PHY register map for the cable environment

15.1.1 Port Status page

Replace Figure 15-2 with the following:

Contents											
Address	0	1	2	3	4	5	6	7			
1000 ₂	AStat		BStat		child	connected	receiveOk	disabled			
1001 ₂	negotiatedSpeed			intEnable	fault	standby Fault	disable Scrambler	BETA_MODE ONLY_PORT			
1010 ₂	dc Connected	maxPortSpeed			IPP	cableSpeed					
1011 ₂	connection Unreliable				beta Mode	tMode	802 Mode	T_MODE PORT			
1100 ₂	portError										
1101 ₂						loop Disable	inStandby	hard Disable			
1110 ₂								long Contend			
1111 ₂											

Figure 15-2—PHY register page 0: Port Status page

Replace Table 15-3 with the following:

Table 15-3—PHY register Port Status page fields (Sheet 1 of 4)

Field	Size	Type	Power reset value	Description
802Mode	1	rc	0	Set for a port if the result of Auto-Negotiation is that the port is operating in 802_mode.
AStat	2	ru	00 ₂	TPA line state for the port (only valid if a DS port): 00 ₂ = invalid 01 ₂ = 1 10 ₂ = 0 11 ₂ = Z

Table 15-3—PHY register Port Status page fields (Sheet 2 of 4)

Field	Size	Type	Power reset value	Description
betaMode	1	ru	0	If equal to 1, the port is operating in Beta mode; equal to 0 otherwise (i.e., when operating in DS mode or when disconnected). If the connected field is 1 and the betaMode field is 0, then the port is operating in DS mode.
BETA_MODE_ONLY_PORT	1	r	see description	Implementation-dependent constant. If equal to 1, the port is not capable of DS mode operation.
BStat	2	ru	00 ₂	TPB line state for the port (only valid if a DS port) (same encoding as the AStat field).
cableSpeed	3	r	see description	Set by implementation-dependent means to the maximum speed at which the cable attached to the port is allowed to operate. If no cable-dependent speed indication is available, then the implementation shall set this variable to the maximum speed of which the port is capable. The encoding is the same as for the maxPortSpeed field.
child	1	ru		If equal to 0, the port is a parent; otherwise it is a child port (or disconnected or disabled). The meaning of this bit is undefined from the time a bus reset is detected until the PHY transitions to state T1: Child Handshake during the tree identification process (see 4.4.2.2).
connected	1	ru	0	If equal to 1, the port is connected and (Beta mode only) operating speed negotiation completed.
connectionUnreliable	1	rcu	0	If equal to 1, a Beta mode speed negotiation has failed or synchronization has failed. A write of 1 to this field resets the value to 0.
dcConnected	1	ru	0	If equal to 1, the port has detected a dc connection to the peer port (by means of a dc connect detect circuit, if implemented).
disableScrambler	1	rw	0	When equal to 1, the port inhibits the operation of the scrambler during transmission of a packet so that transmitted scrambled data are equal in value to unscrambled data. Note that control states and request types continue to be scrambled. Intended for use as a test mode only, not used during normal operation.
disabled	1	rwu	see description	If equal to 1, the port is disabled. The value of this bit subsequent to a power reset is implementation-dependent. If a hardware configuration option is provided, a single option may control the power reset value for all ports. See also the hardDisable field.
fault	1	rcu	0	Set to 1 if an error is detected during a suspend or resume operation; cleared on exit from the suspend state. A write of one to this bit or receipt of the appropriate remote command packet (see 17.3.3.2) shall clear it to zero. When this bit is zeroed, both resume and suspend errors are cleared.

Table 15-3—PHY register Port Status page fields (Sheet 3 of 4)

Field	Size	Type	Power reset value	Description																
hardDisable	1	rw	0	No effect unless the port is disabled. If equal to 1, the port does not maintain connectivity status on an ac connection when disabled. The values of the connected and receiveOk fields are forced to 0. This flag can be used to force renegotiation of the speed of a connection.																
intEnable	1	rw	0	Enables port event interrupts. When set to 1, the PHY shall set the portEvent field to one if any of this port’s Bias (unless the port is disabled), Connected, Disabled, Fault, Standby or standbyFault bits change state.																
localPlugPresent (IPP in the map)	1	ru	see description	Indicates that an external implementation-dependent mechanism has determined that there is at least a physical connection from the local port (maybe not connected at the “far end”). If there is no such mechanism, then this flag shall be set permanently to 1.																
longContendTimer	1	rw	see description	Indicates that long root contend times should be used on this port. Default is 1 for T-mode ports and 0 for all other ports. If software sets this value, then software shall ensure that the same value is used by a port and its peer.																
loopDisable	1	ru	0	Set if the port has been placed in the loopDisable state as part of the loop-free build process (the PHYs at both ends of the connection are active; but if the connection itself were activated, then a loop would exist). Cleared on bus reset and on disconnection.																
maxPortSpeed	3	rw	see description	The maximum speed at which a port is allowed to operate in Beta mode. The encoding is <table><tr><td>000₂</td><td>S100</td></tr><tr><td>001₂</td><td>S200</td></tr><tr><td>010₂</td><td>S400</td></tr><tr><td>011₂</td><td>S800</td></tr><tr><td>100₂</td><td>S1600</td></tr><tr><td>101₂</td><td>S3200</td></tr><tr><td>110₂</td><td>reserved</td></tr><tr><td>111₂</td><td>DS_MODE_ONLY (port is not capable of operating in Beta mode)</td></tr></table> An attempt to write to the register with a value greater than the hardware capability of the port results in the maximum value that the port is capable of being stored in the register. The port uses this register only when a new connection is established in Beta mode. The power reset value is the maximum value that the port is capable of, or the read-only value of 111₂ for a DS_MODE_ONLY port.	000 ₂	S100	001 ₂	S200	010 ₂	S400	011 ₂	S800	100 ₂	S1600	101 ₂	S3200	110 ₂	reserved	111 ₂	DS_MODE_ONLY (port is not capable of operating in Beta mode)
000 ₂	S100																			
001 ₂	S200																			
010 ₂	S400																			
011 ₂	S800																			
100 ₂	S1600																			
101 ₂	S3200																			
110 ₂	reserved																			
111 ₂	DS_MODE_ONLY (port is not capable of operating in Beta mode)																			
negotiatedSpeed	3	ru	000 ₂	Indicates the maximum speed negotiated between this PHY port and its immediately connected port; the encoding is as for the maxPortSpeed field. Set to a speed corresponding to the value of the portSpeed variable. This field is set on connection when in Beta mode, or to value established during self-identification sequence when in DS mode.																

Table 15-3—PHY register Port Status page fields (Sheet 4 of 4)

Field	Size	Type	Power reset value	Description
portError	8	rcu	0	Incremented whenever the port receives an INVALID codeword, unless the value is already 255. Cleared when read (including being read by means of a remote access packet). Intended for use by a single bus-wide diagnostic program.
receiveOk	1	ru	0	In DS mode, indicates the reception of a debounced TpBias signal. In Beta mode, indicates the reception of a continuous electrically valid signal. Note that the receiveOk field is set to false during the time that only connection tones are detected in Beta mode.
inStandby	1	ru	0	Set to 1 if the port in standby (in states P7–P10; see 15.5).
standbyFault	1	rcu	0	Set to 1 if an error is detected during a standby operation and cleared on exit from standby. A write of one to this bit or receipt of the appropriate remote command packet (see 17.3.3.2) shall clear it to zero. When this bit is zeroed, standby errors are cleared.
tMode	1	rc	0	Set for a T_MODE_CAPABLE_PORT or T_MODE_PORT if the result of Auto-Negotiation is that the port is operating in T-mode.
T_MODE_PORT	1	r	see description	Implementation-dependent constant. Set if the port is only capable of operating in T-mode (and, therefore, not in Beta mode or DS mode).

Note that Clause 16 was introduced in IEEE Std 1394b-2002.

16. Data routing, arbitration, and control

16.2 PHY services

16.2.5 PMD services for the PHY

Change 16.2.5.1 and 16.2.5.2 as follows:

16.2.5.1 PMD control request (PMD_CONTROL.request)

The PHY uses the PMD control request service to control the PMD layer. The PMD layer shall service the request immediately upon receipt. No confirmation is sent for this request.

The following parameters are communicated via this service:

- controlRequest. This parameter contains the type of request to be made to the PMD layer. The following values are defined for this parameter:
 - PMD_CROSSOVER. A TPA-TPB crossover is implemented (see 12.4.5).
 - PMD_NO_CROSSOVER. Any TPA-TPB crossover is removed (see 12.4.5).
 - PMD_TONE_ON. The PMD layer is instructed to generate a tone (see 9.6.1).
 - PMD_TONE_OFF. The PMD layer is instructed to cease generating a tone, remove the signalling bias voltage, and set the port transmitters to high impedance.
 - PMD_SELECT_BPORT. The Beta port PMD layer is selected for Beta mode input and output (I/O).
 - PMD_SELECT_DS_PORT. The DS port PMD layer is selected for data, arbitration line states, and common-mode speed signals.
 - PMD_UNSELECT_PORT. Control of port signalling is removed from the T-mode port, Beta port, or DS port (as appropriate). A T-mode port will maintain connectivity status.
 - PMD_START_NEGOTIATE. The PMD layer is instructed to perform T-mode/Ethernet negotiation.
 - PMD_SELECT_TPORT. The PMD layer is instructed to use the port for T-mode I/O (set only while tportOn is TRUE; unset during suspend, etc.) The port will synchronize with its peer and return the status of PMD_TPORT_OK when synchronization has been established.
 - PMD_TPORT_OFF. The PMD layer is disabled.
- speed. (Conditional, present only when the controlRequest parameter is PMD_SELECT_BPORT.) This parameter indicates the operating speed of the port.

16.2.5.2 PMD status request (PMD_STATUS.request) and confirmation (PMD_STATUS.confirmation)

The PHY uses the PMD status request and confirmation service pair to request status from the PMD layer. The PMD layer shall service the request immediately upon receipt and respond with the PMD status confirmation.

NOTE—The C code uses a unified PMD_STATUS_request, which combines the functions of this request and the corresponding confirmation.

The following parameters are communicated by the PMD confirmation:

- PMD_BIAS_DETECT. This parameter contains the undebounced output of the PMD bias comparator.
- PMD_CONNECT_DETECT. This parameter contains the undebounced PMD output of connect detect comparator.
- PMD_LOCAL_PLUG_PRESENT. When ac-coupled (possibly via, e.g., an optical transceiver), this parameter indicates that an external implementation-dependent mechanism has determined that at least a physical connection exists from the local node to a cable (although a connection may not exist to the peer port). This parameter is used to avoid performing connection toning if no connection definitely exists. If no such mechanism exists, then this parameter shall be set permanently to TRUE.
- PMD_SIGNAL_DETECT. The PMD layer sets this parameter if it detects a signal as specified by the signal detect specification of the specific PMD layer.
- PMD_AUTOCROSSOVER_SUPPORTED. The PMD layer sets this parameter if the crossover control requests are supported.
- PMD_NEGOTIATION_ACTIVE. The PMD layer sets this parameter while it is carrying out T-mode/Ethernet negotiation.
- PMD_TPORT_FW. The PMD layer sets this parameter if Auto-Negotiation has established T-mode.
- PMD_TPORT_802. The PMD layer sets this parameter if Auto-Negotiation has established operation according to IEEE 802.3 higher layer specifications.
- PMD_TPORT_OK. The PMD layer sets this parameter if the connection is synchronized (T-mode only) and ready for normal control symbol, arbitration symbol, and data exchange.

16.3 PHY facilities

Replace 16.3.4 (including Table 16-9) with the following:

16.3.4 Cable interface timing constants

NOTE—This subclause defines new values and changes some existing constants from which the configuration and timing of the PHY in the cable environment may be derived.

Many of the values in Table 16-9 are required for hybrid bus operation. These values apply to buses where at least one of the connections between nodes is done with DS ports. In Table 16-9, Legacy packet means both Legacy packets on a Beta connection and any packet on a DS connection. A few constants are required only for border nodes.

Repeating ports are required to be designed to account for clock frequency and phase differences and still guarantee relevant times from Table 16-9.

Table 16-9—Cable interface timing constants (Sheet 1 of 3)

Timing constant	Minimum	Maximum	Comment
ARB_RESPONSE_DELAY	Maximum PHY_DELAY less 0.08 μ s	See comment	Delay through a PHY from the start of reception of an arbitration line state to the start of transmission of the associated arbitration line state at all transmitting ports. Arbitration shall be repeated by the PHY at least as fast as clocked data. Applies to Legacy ports and to the control states at the start and end of packets.
ARB_SPEED_SIGNAL_START	−0.02 μ s	—	(DS ports only.) Time delay between a DS transmitting port signalling TX_DATA_PREFIX and the same port transmitting a speed signal for either an unconcatenated packet or the first packet in a concatenated sequence.
BASE_RATE	98.294 Mb/s	98.314 Mb/s	Base bit rate (98.304 Mb/s $\pm$ 100 ppm)
BOSS_RESTART_TIME	10 μ s	11 μ s	Time allowed for a subaction to complete and to ensure that tokens are seen consistently on the bus (used only if a node does not use the Legacy gap count).
CM_MIN_IDLE_TIME	1.070 μ s	—	Idle time not in the isochronous interval at proxy_root with a Legacy link.
CONCATENATION_PREFIX_TIME	0.16 μ s	—	For concatenated Legacy packets, the time a transmitting port shall signal DATA_PREFIX between the end of clocked data for one packet and the start of speed signaling time for the next.
CONFIG_TIMEOUT	166.6 μ s	166.9 μ s	Loop detect time. ($\sim 16384/\text{BASE_RATE}$)
DATA_END_TIME	0.24 μ s	0.26 μ s	End of packet signal time for a Legacy packet. ($\sim 24/\text{BASE_RATE}$)
DATA_PREFIX_HOLD	0.04 μ s	—	At a transmitting port, the time between the end of speed signalling (when present) and the start of clocked data for a Legacy packet.
FORCE_ROOT_TIMEOUT	83.32 μ s	83.34 μ s	Time to wait in T0: Tree ID Start state (see Figure 16-8) before acknowledging RX_PARENT_NOTIFY. ($\sim 8192/\text{BASE_RATE}$)
LONG_BOSS_RESTART_TIME	83.32 μ s	83.34 μ s	Time allowed for a subaction to complete (used only if a node does not use the Legacy gap count and if large diameter networks are supported)
LONG_ROOT_CONTENTEND_FAST	3.58 μ s	3.59 μ s	Time to wait in state T3: Root Contention if longContentTimer is selected and the random bit is zero, as described in 16.4.6. ($\sim 352/\text{BASE_RATE}$)

Table 16-9—Cable interface timing constants (Sheet 2 of 3)

Timing constant	Minimum	Maximum	Comment
LONG_ROOT_CONTENTEND_SLOW	7.16 μ s	7.18 μ s	Time to wait in state T3: Root Contention if longContentendTimer is selected and the random bit is one, as described in 16.4.6. ($\sim 704/\text{BASE_RATE}$).
MAX_ARB_STATE_TIME	200 μ s	400 μ s	Maximum time in any state (before a bus reset is initiated) except A0: Idle state (during the time that idleArbStateTimeout is false), T0: Tree ID Start state, or a state that exits after an explicit timeout.
MAX_BETA_TIME	83.324 μ s	104.5 μ s	(Senior border node only.) Maximum time a senior border node shall wait after the beginning of any Beta traffic for a Legacy packet before sending a BORDER request. (When the BORDER request is granted, the senior border node shall send a Legacy packet if still required to free the various Legacy clouds. (Minimum is $8192/\text{BASE_RATE}$.)
MAX_BUS_HOLD	—	1.63 μ s	Maximum time an originating node may transmit DATA_PREFIX between concatenated packets. The link shall ensure that this time is not exceeded.
MAX_DATA_TIME	—	84.34 μ s	Maximum time that clocked data may be transmitted continuously. If this limit is exceeded, unpredictable behavior may result.
MIN_DATA_PREFIX	0.14 μ s	—	Time a port transmitting a Legacy packet is required to signal DATA_PREFIX prior to clocked data for either an unconcatenated packet or the first packet in a concatenated sequence.
MIN_DELETABLE_SYMBOL_TIME	0.04 μ s	—	Minimum time that an originating port is required to insert deletable symbols. ($\sim 4/\text{BASE_RATE}$)
MIN_IDLE_TIME	0.04 μ s	—	Minimum idle time (not including speed signalling on a Legacy port) after transmission of DATA_END for a Legacy packet at either an originating or repeating port. ($\sim 4/\text{BASE_RATE}$)
MIN_PACKET_SEPARATION	0.34 μ s	—	Minimum time that an originating port is required to signal DATA_PREFIX between concatenated Legacy packets. ($\sim 34/\text{BASE_RATE}$)
NOMINAL_CYCLE_TIME	124.988 μ s	125.013 μ s	Average time between the start of one isochronous period and the next. ($125 \mu\text{s} \pm 100 \text{ ppm}$)

Table 16-9—Cable interface timing constants (Sheet 3 of 3)

Timing constant	Minimum	Maximum	Comment
PHY_DELAY	0.06 μ s if PHY_DELAY = 144 ns See comment	See PHY registers (Clause 15)	Measured from the receipt of the first data bit to its retransmission by the repeating port(s) for all combinations of ports and speeds. Best-case repeater data delay has a fixed minimum. If PHY_DELAY is reported in the PHY registers as being greater than 144 ns, then software may use the value (reported PHY_DELAY – reported JITTER) if this value is greater.
RESET_TIME	166.6 μ s	166.7 μ s	Reset hold time. ($\sim 16384/\text{BASE_RATE}$)
RESET_WAIT	0.16 μ s		Reset wait delta time. ($\sim 16/\text{BASE_RATE}$)
RESPONSE_TIME	0.04 μ s	PHY_DELAY + 0.1 μ s	1) Idle time at all ports of the responding node, measured at the cable connector, from the end of DATA_END or DATA_END that follows a PHY packet or primary packet to the start of the next arbitration line state, DATA_PREFIX, DISABLE_NOTIFY, or SUSPEND. Applies to responding node. 2) Time between receipt of LTP value change and corresponding LTS transmit.
ROOT_CONTEND_FAST	1.62 μ s	1.63 μ s	Time to wait in state T3: Root Contention if the random bit is zero, as described in 16.4.6. ($\sim 160/\text{BASE_RATE}$)
ROOT_CONTEND_SLOW	3.25 μ s	3.26 μ s	Time to wait in state T3: Root Contention if the random bit is one, as described in 16.4.6. ($\sim 320/\text{BASE_RATE}$)
SHORT_RESET_TIME	1.26 μ s	1.40 μ s	Short reset hold time. ($\sim 128/\text{BASE_RATE}$)
SID_SPEED_SIGNAL_START	–0.02 μ s	0.02 μ s	(DS ports only.) Time between a child DS port signalling TX_IDENT_DONE and the same port sending its speed capability signal.
SPEED_SIGNAL_LENGTH	0.10 μ s	0.12 μ s	Duration of a valid time that speed signal output is driven ($\sim 10/\text{BASE_RATE}$) for Legacy packets.

16.4 Cable PHY operation

Replace 16.4.1 (including Table 16-10) with the following:

16.4.1 C code functions and variables

The C code functions and variables used in the arbitration state machine are described in Table 16-10.

Table 16-10—C code functions and variables (Sheet 1 of 3)

Function or variable name	Description
active[port]	Indicates that the port is in the P2: Active state.
allChildPortsIdentified	Set if all child ports have been identified.
arbOk()	True if OK to request the bus.
arbPowerReset()	Power reset initialization routine.
arbReset	True to request an arbitration reset token to be sent.
arbTimer	Timer for arbitration state machines.
betaMode[port]	Set if the port has determined that it is operating in Beta mode; otherwise, unset (i.e., when operating in DS mode or when in P0: Disconnected state). This variable is maintained as a read-only bit in the port register map.
betaPortRequest	True if a request on a port is to be granted.
child[port]	True if the port is a child or not active.
childHandshakeComplete()	True once all active children are in S0: Self-ID Start state.
child_ID_complete[port]	True when self-identification sequence is complete on a specified child port.
children	Number of child ports.
concatenatedPacket	True if a concatenated packet is being received.
contendTime	Amount of time to wait during root contention.
convertedRequest	True if a request has been converted from a Beta request into a Legacy request to be forwarded on a DS port.
dataComing()	True if DATA_PREFIX, SPEED, DATA_NULL, or CYCLE_START token is detected on a port.
dataComingOn(port)	True if DATA_PREFIX, SPEED, DATA_NULL, or CYCLE_START token is detected on the specified port.
endOfPacket	Set when reception of packet is complete.
flyByOk	True when fly-by concatenation permitted.
forceRoot	A value of TRUE modifies the PHY's tree identification behavior and increases the likelihood that the node becomes root (see 4.4.2.2). If only one node on a bus has forceRoot set to TRUE, that node is guaranteed to become the root.
fromA0	True on entry to A2: Grant state if the previous arbitration state was A0: Idle state.
gapRepeatActions(outOfPacket)	Repeats gap tokens.
grantSelf	True if can grant to self.
ibr	Set when a long bus reset is needed.
idleReceivePort	True if port goes idle after receiving self-ID packets.
immediateRequest	True if immediate request from link.