

IEEE Std 1394b™-2002
(Amendment to IEEE Std 1394™-1995)

1394b™

IEEE Standard for a High-Performance Serial Bus—Amendment 2

IEEE Computer Society

Sponsored by the
Microprocessor and Microcomputer Standards Committee



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

14 December 2002

Print: SH94986
PDF: SS94986

Recognized as an
American National Standard (ANSI)

IEEE Std 1394b™-2002
(Amendment to
IEEE Std 1394™-1995)

IEEE Standard for a High-Performance Serial Bus—Amendment 2

Sponsor

Microprocessor and Microcomputer Standards Committee
of the
IEEE Computer Society

Approved 1 August 2002

American National Standards Institute

Approved 20 March 2002

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 © 2002 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 14 December 2002. Printed in the United States of America.

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

Print: ISBN 0-7381-3253-5 SH94986
PDF: ISBN 0-7381-3254-3 SS94986

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.

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
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

Note: 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 for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

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 1394b-2002, IEEE Standard for a High-Performance Serial Bus—Amendment 2.)

Work on IEEE Std 1394b-2002 was begun in the winter of 1997 at a meeting of the 1394TA in Eindhoven. By that time, it had become evident that IEEE Std 1394-1995 was going to be widely deployed and that many devices, especially new, digital consumer products, were going to have IEEE 1394 as the primary external interface. In the first meetings of this group, considerable sentiment existed for broadening the number and types of devices that would be able to use the IEEE 1394 interface and thereby making the interface more valuable to the end user. While attempting to broaden the scope of the IEEE 1394 interface, two barriers were discovered that made wider deployment difficult: the interface was constrained to operate over a fairly short distance and it could not handle higher data rates.

After some preliminary investigations, the group concluded that these problems were best solved by a change in coding. While the data-strobe (DS) coding used in IEEE Std 1394-1995 was a simple, self-clocking scheme, the fact that it is not dc balanced made it impractical to use over longer distances. Furthermore, accumulated skew on a long-distance connection makes it hard to maintain the timing relationships between the data and the strobe lines, especially at higher data rates. The group rapidly converged on the notion of using a variant of 8B/10B coding developed by IBM for the longer distances and higher speeds defined in IEEE Std 1394b-2002. Where the connection between devices was copper cable of 5 m or less, some or all of the ports on a new physical layer (PHY) developed to support this standard could be able to signal using either DS or the new signaling scheme (dubbed *Beta mode*). These *bilingual* ports would be able to select the optimum signaling method for the connection and thereby be able to fully interoperate with existing devices that are compliant with IEEE Std 1394-1995 and IEEE Std 1394a™-2000.

The new signalling system provided a route to solving a second major problem, i.e., the lack of scalability of the IEEE 1394 arbitration scheme as signaling rates are increased. The use of 8B/10B encoding allows the transmission of data to be overlapped with the transmission of arbitration signals in the reverse direction. The use of overlapped arbitration is extended by one level of pipelining, with the result that in a purely IEEE 1394b bus the need for arbitration gaps is entirely eliminated. In fact, on a bus with all connections operating in Beta mode, the setting of gap count has no utility as the bus is completely self-timed.

As work progressed, the group became interested in applying the new coding scheme to a large variety of interconnect media including plastic optical fiber (POF), glass optical fiber (GOF), and Category 5 (CAT-5) unshielded twisted pair (UTP). This goal greatly expanded the scope of the work and made it necessary to divide the IEEE P1394b Working Group into various task groups. The leaders of these groups were largely responsible for the development of IEEE Std 1394b-2002. Because of the length of the project, some of the task groups had more than one leader. The task groups, their leaders, and their contributions are as follows:

- *Glass Fiber*. Colin Whitby-Strevens. This group adopted the work from the Fibre Channel specification for use in IEEE Std 1394b-2002. The jitter work done in this group was adapted for use in all the media.
- *Plastic Fiber*. Taka Fujimori, Shuntaro Yamazaki, Victoria K. M. Teng, and Kazuki Nakamura. This group actually developed a standard around a single connector that allowed interoperation and intermating of both plastic fiber or large-diameter hard polymer clad fiber (HPCF). This connector can be used at a speed of S100.
- *UTP5*. Colin Whitby-Strevens and Alistair Coles. This group built on work done in the 100BaseT standard with extensive changes to simplify the coding and transmission over CAT-5 UTP.

- *Standard Electrical*. Eric Hannah. This group provided the electrical characteristics for the base-line PHY. This required a great deal of simulation work by Eric Hannah in support of the Cable & Connector task group.
- *Start-up Protocol*. Colin Whitby-Strevens. This group created the mechanisms that allow a B PHY to determine whether it is attached to another B PHY and to select the signaling frequency before the connection is completed. Peng Zhang and Jim Nave proposed the speed signaling scheme using a toning mechanism. This was enhanced by Colin Whitby-Strevens to deal with all the connectivity aspects of B ports. The loop-breaking protocol was developed by Jerry Hauck and David Wooten.
- *Cable & Connector*. Max Bassler. This group originally tried to characterize the 4- and 6-circuit connectors used in IEEE Std 1394-1995 and IEEE Std 1394a-2000 for use at the higher speeds defined in IEEE Std 1394b-2002. This work was abandoned after two years, and a new, small form-factor connector was developed by Dave Brunker for use on devices capable of Beta-mode signaling.
- *Protocol Accelerations*. David LaFollette, Alistair Coles, and Mike Teener. This group developed the bus owner/supervisor/selector (BOSS) arbitration scheme that is the basis for all of the arbitration and control. The original concepts were proposed by David LaFollette and Jerry Hauck. Their work was supplemented by contributions from Alistair Coles, Mike Teener, Jim Skidmore, and David Wooten.
- *Port Logic*. Alistair Coles. This group developed the coding and scrambling methods used for Beta ports. Alistair Coles and Eric Deliot developed the single character control codes and arbitration tokens. This group also selected the scrambling method for the symbols. The method of dealing with deletable symbols was adapted from the Fibre Channel specification by Jerry Hauck and Mike Teener.
- *Simulation*. Jerry Hauck. This group did computer simulations of the various pieces of the B PHY, but most of the simulation was mental and took place in Jerry Hauck's mind. This turned out to be a much more efficient method of finding problems than a computer simulation.
- *Low-power*. Richard Churchill and Steve Bard. This group developed the standby-restore protocol.
- *PHY-link/PIL-FOP*. Martin Sodos, Sean Killeen, and David Wooten. This group developed the source synchronous version of the PHY-link interface and the PIL-FOP interface. The point-to-point (P2P) packet protocol for sending in-band status and register information between the PHY that is integrated into the link (PIL) and the fan-out PHY (FOP) was developed by Jerry Hauck and David Wooten.
- *C code*. Colin Whitby-Strevens. Although not done as a formal task group, the work that Colin Whitby-Strevens did in developing and maintaining the C code for the standard bears special recognition.

The IEEE P1394b Working Group first met under the chairmanship of Mike Teener in January 1997. David Wooten became chair within a few months, and the working group continued to meet approximately every month until the editorial work was completed in October of 1999 at which time the working group voted to go to ballot. The proposed standard was balloted in March and April of 2000 with over 80% of the votes in favor. A Ballot Response Committee (BRC) was formed to review the comments. Those comments pointed out many significant omissions and errors in the draft that the BRC corrected over the next seven months of meetings. The BRC completed its work on this standard in February of 2001.

The ballot for the February 2001 version of the standard passed but with comments that pointed out some flaws that would need to be corrected before the standard could be released. The BRC met several times from June to September and completed work on a new version of the standard in September of 2001.

Patent notice

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 all patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

The patent holder has, however, filed a statement of assurance that it will grant a license under these rights without compensation or under reasonable rates and nondiscriminatory, reasonable terms and conditions to all applicants desiring to obtain such a license. The IEEE makes no representation as to the reasonableness of rates and/or terms and conditions of the license agreement offered by the patent holder. Contact information may be obtained from the IEEE Standards Department.

Participants

The following is a list of major participants in the IEEE P1394b working group (those that attended at least three working group meetings.)

David R. Wooten, *Chair*

Michael Johas Teener, *Vice Chair*

Steve Bard, *Secretary*

Eric Hannah, *Editor*

Bill Anderson
Tatsuya Arai
Oleg Awsienko
Richard Baker
David Barnum
Max Bassler
Charles Brill
Dave Brown
Mike Brown
Dave Brunker
Jim Busse
Don Chambers
Dao-Long Chen
Richard Churchill
Dan Colegrove
Alistair Coles
Mike Coletta
Eric Deliot
Chris Dorsey
Firooz Farhoomand
Stephen Finch
Mike Fogg
Tony Foster
John Fuller
Nobuo Furuya
Dave Gampell
Bob Gannon
Mike Gardner
Jerry Hauck
Keith Heilmann
John Hill
Daisuke Hiraoka
Jack Hollins
Derek Imschweiler

Tatsuo Inoue
David Instone
David James
David Johnson
Tom Jones
Prashant Kanhere
Marcus Kellerman
Al Kelley
Sean Killeen
Akihito Kuwabara
Dave LaFollette
Farrukh Latif
Thang Le
Paul Levy
Francesco Liburdi
Robert Liu
John Lopata
Gerald Marazas
Jun-ichi Matsuda
Edward McDonnell
Daniel Meirsman
Jack Merrow
Takatoshi Mizoguchi
Reza Moattar
Palanisamy Mohanraj
Cyrus Momeni
Karl Nakamura
James Nave
Jim Nelson
Richard Nesin
Bill Northey
Takayuki Nyu
Ozay Oktay
Bijit Patel

James Piccione
Bill Prouty
Dennis Rehm
Kyoza Saito
Tomoki Saito
Brad Saunders
Dick Scheel
D. C. Sessions
Masood Shariff
Robbie Shergill
Jim Skidmore
David Smith
Michael Smith
John Smolka
Ron Soderstrom
Martin Sodos
John Ta
Ju-Ching Tang
Ken Taylor
Peter Teng
Victoria Teng
Tom Thatcher
David Thompson
Toru Ueda
Sushant Verman
Hirosha Wakai
Kenji Watanabe
Yuji Watanabe
Colin Whitby-Stevens
Shuntaro Yamazaki
Niwa Yoshikatsu
Len Young
Patrick Yu
Peng Zhang

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

Eric Anderson
Steven R. Bard
Brian D. Batchelder
David Brearley
Keith Chow
Robert S. Cram
James R. Davis
Dante Del Corso
Chris S. Dorsey
Sourav K. Dutta
Roger D. Edwards
Firooz Farhoomand
Gordon Force
John Fuller
Eric Hannah
Jerry Hauck
Burke Henahan
Clay E. Hudgins

Peter Johansson
David K. Johnson
Sean Killeen
Thomas M. Kurihara
Lawrence Lamers
Steven Larky
Gerald A. Marazas
Joseph R. Marshall
Gene E. Milligan
Kiyoshi Miura
Robert Mortonson
Klaus-Dieter Mueller
Atsushi Nakamura
William A. Northey
Farrell Ostler
Dan Paley
Gary S. Robinson
Gordon Robinson

John Rynearson
Bradley N. Saunders
Thomas J. Schaal
Larry Sheu
James M. Skidmore
Larry Stein
Junichi Takeuchi
Joseph Tardo
Michael D. Teener
David W. Thompson
Clarence M. Weaver, Jr.
Colin Whitby-Stevens
David R. Wooten
Don Wright
Roy Yasoshima
Tadahiro Yoshida
Patrick W. Yu
Oren Yuen

When the IEEE-SA Standards Board approved this standard on 20 March 2002, it had the following membership:

James T. Carlo, *Chair*

James H. Gurney, *Vice Chair*

Judith Gorman, *Secretary*

Sid Bennett
H. Stephen Berger
Clyde R. Camp
Richard DeBlasio
Harold E. Epstein
Julian Forster*
Howard M. Frazier
Toshio Fukuda

Arnold M. Greenspan
James H. Gurney
Raymond Hapeman
Donald M. Heirman
Richard H. Hulett
Lowell G. Johnson
Joseph L. Koepfinger*

Peter H. Lips
Nader Mehravari
Daleep C. Mohla
William J. Moylan
Malcolm V. Thaden
Geoffrey O. Thompson
Howard L. Wolfman
Don Wright

*Member Emeritus

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

Alan Cookson, *NIST Representative*

Satish K. Aggarwal, *NRC Representative*

Catherine Berger

IEEE Standards Project Editor

Contents

0. Overview	1
0.1 Scope	1
0.2 Purpose	2
0.3 Document organization	2
1. Overview	3
1.2 References	3
1.5 Service model	4
1.6 Document notation	5
2. Definitions and abbreviations	13
2.1 Conformance	13
2.2 Technical	13
2.3 Acronyms and abbreviations	21
3. Summary description	23
3.10 New features of IEEE Std 1394b-2002	23
4. Cable PHY specification	35
9. Short-haul copper PMD electrical specification	77
9.1 Interfaces	78
9.2 Transmitter electrical specifications	78
9.3 Receiver electrical specifications	81
9.4 Electrical measurements	82
9.5 DC biasing	83
9.6 Toning and signal detect	83
9.7 Jitter	86
10. GOF PMD specification	89
10.1 PMD block diagram	89
10.2 PMD to MDI optical specifications	90
10.3 Transmitter optical specifications	91
10.4 Receiver optical specifications	91
10.5 Worst-case connection optical power budget and penalties (informative)	92
10.6 Optical jitter specifications	92
10.7 Optical measurement requirements	94
10.8 CPR measurement	96
10.9 Optical connection cabling model	96
10.10 Optical connection	97
10.11 Fiber launch conditions: overfilled launch (OLF)	98
11. PMD specification of fiber media with PN connector	99
11.1 Scope	99
11.2 PMD block diagram	100

11.3 Cables	100
11.4 Connector	101
11.5 Connector and cable assembly performance criteria	101
11.6 Optical fiber interface	102
11.7 Optical jitter specifications	103
11.8 Permitted number of segments (informative)	104
12. Cat-5 UTP PMD specification	107
12.1 Overview	107
12.2 PMD block diagram	108
12.3 Operation of CAT-5 connections	108
12.4 Media specification.....	108
12.5 PMD electrical specifications	110
12.6 PMD implementation (informative)	114
13. Beta port specification.....	115
13.1 Overview	115
13.2 Port functions	116
13.3 Beta port operation	133
13.4 Beta port state machines	140
14. Connection management.....	145
14.1 Overview	145
14.2 Port characteristics	146
14.3 Functions, variables, and constants	147
14.4 Port controller.....	149
14.5 Port connection manager state machine.....	149
14.6 Standby.....	154
14.7 Loop prevention.....	155
14.8 Connection management	161
15. PHY register map.....	165
15.1 PHY register map for the cable environment.....	165
15.2 Integrated link and PHY	172
16. Data routing, arbitration, and control	173
16.1 Overview	173
16.2 PHY services	174
16.3 PHY facilities	184
16.4 Cable PHY operation.....	196
17. B PHY- link interface (parallel)	217
17.1 B PHY-link interface characteristics.....	217
17.2 PHY-link interface signals	218
17.3 Interface initialization, reset, and disable	220
17.4 Link-on and interrupt indications	224
17.5 Link requests and notifications	225
17.6 Interface data transfers	231

17.7 Format of received and transmitted data	238
17.8 Status transfers and notifications from the PHY	240
17.9 Delays affecting interoperability of PHYs and links	243
17.10 Legacy link support	244
17.11 Electrical characteristics	245
18. PIL-FOP serial interface	253
18.1 Operating model	253
18.2 PIL-FOP connection management	254
18.3 Serial Bus configuration request types not carried over the PIL-FOP interface	256
18.4 P2P packet protocol	256
19. C code	259
19.1 Common declarations and functions	259
19.2 Connection management routines	273
19.3 Port state machine actions	295
19.4 Border arbitration actions and conditions	314
19.5 Border arbitration	349
Annex K (informative) Serial Bus cable test procedures	361
Annex O (normative) Jitter measurements	365
Annex P (informative) Connection status change	367
Annex Q (informative) Bibliography	369

IEEE Standard for a High-Performance Serial Bus—Amendment 2

EDITORIAL NOTE—IEEE Std 1394b™-2002 is based on the current edition of IEEE Std 1394™-1995 and the approved amendment IEEE Std 1394a™-2000. The editing instructions define how to merge the material contained here into this base document set to form the new comprehensive standard as created by the addition of IEEE Std 1394b-2002.

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 ~~strike through~~ (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

IEEE Std 1394b-2002 is a full-use standard whose scope is to provide an amendment to IEEE Std 1394-1995 and a companion to IEEE Std 1394a-2000. This amendment defines features and mechanisms that provide gigabit speed extensions in a backwards compatible fashion and the ability to signal over single hop distances of up to 100 m.

The following are included in IEEE Std 1394b-2002:

- a) Cables and connectors for gigabit signaling and long distance (up to 100 m);
- b) Detection of physical loops in the bus topology and their subsequent resolution by selective disabling of physical layer (PHY) port(s);
- c) The same circuits may be used to transmit either data-strobe (DS) signals or 8B/10B-encoded signals;
- d) Extension of the PHY-link interface to higher data rates over an 8 bit parallel bus;
- e) Extension of the PHY-link interface to higher data rates over a bit-serial bus;
- f) Definition of high-speed bit protocols to handle various bus arbitration signals and the transport of slower data packets by bit-stuffed, high-speed packets;
- g) Protocols for signal speed matching at connect time;
- h) Port initialization protocols suitable for use over ac-coupled connections; and
- i) Various informative annexes to discuss testing and compliance procedures for gigabit connections.

The preceding are arranged in no particular order.

0.2 Purpose

Technology advances have made gigabit signaling a feasible and attractive extension to the baseline standard, IEEE Std 1394-1995. In particular, the needs of data storage and home backbones dictate a higher speed and longer distance interconnect bus than can be provided by IEEE Std 1394-1995 and IEEE Std 1394a-2000.

0.3 Document organization

IEEE Std 1394b-2002 contains this overview, a list of definitions, an informative summary description, clauses of technical specification, and application annexes. The new reader should read the informative summary and the clauses that precede it before the remainder of the document.

1. Overview

1.2 References

Insert the following references into 1.2 in IEEE Std 1394-1995, in alphabetical order, and renumber the footnotes:

The standards named in this clause contain provisions that, through reference in this text, constitute provisions of IEEE Std 1394b-2002. At the time of publication, the editions indicated were valid. All standards are subject to revision; parties to agreements based on IEEE Std 1394b-2002 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below.

AF-PHY-0015.000 (September, 1994), ATM Physical Medium Dependent Interface Specification for 155 Mb/s over Twisted Pair Cable.¹

ANSI X3.230-1994 (Reaff 2001), American National Standard for Information Technology—Fibre Channel—Physical and Signaling Interface (FC-PH).²

ANSI X3.263-1995 (Reaff 2000), American National Standard for Information Technology—Fibre Distributed Data Interface (FDDI)—Token Ring Twisted Pair Physical Layer Medium Dependent (TP-PMD).

ANSI Y14.2M-1992 (Reaff 1998), Line Conventions and Lettering.

ANSI Y14.5M-1994 (Reaff 1999), Dimensioning and Tolerancing.

ANSI/EIA 364-D-7-01, Electrical Connector Test Procedures Including Environmental Classifications.³

ANSI/EIA 364-06B-00, Contact Resistance Test Procedure for Electrical Connectors.

ANSI/EIA 364-09C-99, Durability Test Procedure for Electrical Connectors and Contacts.

ANSI/EIA 364-13B-98, Mating and Unmating Forces Test Procedures for Electrical Connectors.

ANSI/EIA 364-17B-99, Temperature Life With or Without Electrical Load Test Procedure for Electrical Connectors and Sockets.

ANSI/EIA 364-18A-84, Visual and Dimensional Inspection for Electrical Connectors.

ANSI/EIA 364-20B-99, Withstanding Voltage Test Procedure for Electrical Connectors, Sockets and Coaxial Contacts.

ANSI/EIA 364-21C-00, Insulation Resistance Test Procedure for Electrical Connectors, Sockets and Coaxial Contacts.

ANSI/EIA 364-23A-85, Low Level Contact Resistance Test Procedure for Electrical Connectors and Sockets.

ANSI/EIA 364-27B-96, Mechanical Shock (Specified Pulse) Test Procedure for Electrical Connectors.

ANSI/EIA 364-28D-99, Vibration Test Procedure for Electrical Connectors and Sockets.

ANSI/EIA 364-31B-00, Humidity Test Procedure for Electrical Connectors and Sockets.

ANSI/EIA 364-32C-00, Thermal Shock (Temperature Cycling) Test Procedure for Electrical Connectors and Sockets.

ANSI/EIA 364-38B-99, Cable Pull-Out Test Procedure for Electrical Connectors.

ANSI/EIA 364-41C-99, Cable Flexing Test Procedure for Electrical Connectors.

ANSI/EIA 364-46A-98, Microsecond Discontinuity Test Procedure for Electrical Connectors, Contacts and Sockets.

ANSI/EIA 364-65A-98, Mixed Flowing Gas.

ANSI/EIA 364-102-98, Rise Time Degradation Test Procedure for Electrical Connectors, Sockets, Cable Assemblies or Interconnection Systems.

ANSI/EIA 364-103-99, Propagation Delay Test Procedure for Electrical Connectors, Sockets, Cable Assemblies or Interconnection Systems.

ANSI/TIA/EIA-455-54B-01, Mode Scrambler Requirements for Overfilled Launching Conditions to Multimode Fibers.

¹ATM Forum publications are available from the ATM Forum, 2570 West El Camino Real, Suite 304, Mountain View, CA 94040-1313. They may also be downloaded from <http://www.atmforum.com>.

²ANSI publications are available from Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036-8002, USA.

³EIA publications are available from Global Engineering Documents, 15 Inverness Way East, Englewood, Colorado 80112, USA (<http://global.ihs.com/>).

ANSI/TIA/EIA-455-95A-00, Absolute Optical Power Test for Optical Fibers and Cables.

ANSI/TIA/EIA-455-127-91, Spectral Characterization of Multimode Laser Diodes.

ANSI/TIA/EIA-526-14A-98, Optical Power Loss Measurement of Installed Multimode Fiber Cable Plant.

ANSI/TIA/EIA-568-A-00, Commercial Building Telecommunications Cabling Standard.

ANSI/TIA/EIA-604-93 (Reaff 2000), Fiber Optic Connector Intermateability Standards (FOCIS).

ANSI/TIA/EIA TSB67-1995, Transmission Performance Specifications for Field Testing of Unshielded Twisted-Pair Cabling Systems.

IEC 60603-7 (1996-11), Connectors for frequencies below 3 MHz for use with printed boards—Part 7: Detail specification for connectors, 8-way, including fixed and free connectors with common mating features, with assessed quality.⁴

IEC 60793-1-1 (1999-02), Optical fibres—Part 1-1: Generic specification—General.

IEC 60793-1-41 (2001-07), Optical fibres—Part 1-41: Measurement methods and test procedures—Bandwidth.

IEC 60793-2 (2001-10), Optical fibres—Part 2: Product specifications.

IEC 61000-4-2 (2001) Edition 1.2, Electromagnetic compatibility (EMC)—Part 4-2: Testing and measurement techniques—Electrostatic discharge immunity test.

ISO/IEC 11801:2000, Information technology—Generic cabling for customer premises.⁵

IEEE Std 1394b-2002 shall also be used in conjunction with the following publications under development. When approved as a standard, the approved version shall apply.

IEEE Std 1212™-2001, IEEE Standard for a Control and Status Registers (CSR) Architecture for Microcomputer Buses.⁶

NCITS TR-25-1999, Information Technology—Fibre Channel—Methodologies for Jitter Specification.⁷

Replace 1.5 in IEEE Std 1394-1995 with the following:

1.5 Service model

IEEE Std 1394-1995 and this amendment both use a protocol model with multiple layers. Each layer provides services to the next higher layer and to Serial Bus management. These services are abstractions of a possible implementation; an actual implementation may be significantly different and still meet all the requirements. The method by which these services are communicated between the layers is not defined by this amendment. Four types of service are defined by IEEE this standard. They are as follows:

⁴IEC publications are available from the Sales Department of the International Electrotechnical Commission, Case Postale 131, 3, rue de Varembe, CH-1211, Genève 20, Switzerland/Suisse (<http://www.iec.ch/>). IEC publications are also available in the United States from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA.

⁵ISO/IEC publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembe, CH-1211, Genève 20, Switzerland/Suisse. ISO publications are also available in the United States from the American National Standards Institute.

⁶IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA (<http://standards.ieee.org/>).

⁷NCITS T11.2 working documents are available at <http://www.t11.org>.

- a) *Request service.* A request service is a communication from a layer to a lower or adjacent layer to request some action. A request may also communicate parameters that may or may not be associated with an action. A request may or may not be confirmed. A data transfer request usually triggers a corresponding indication on peer node(s). (Because broadcast addressing is supported on Serial Bus, it is possible for the request to trigger a corresponding indication on multiple nodes.)
- b) *Indication service.* An indication service is a communication from a layer to a higher or adjacent layer to indicate a change of state or other event detected by the originating layer. An indication may also communicate parameters that are associated with the change of state or event. Indications are not necessarily triggered by requests; an indication may or may not be responded to by a response. A data transfer indication is originally caused by a corresponding request on a peer node.
- c) *Response service.* A response service is a communication from a layer to a lower or adjacent layer in response to an indication; a response is always associated with an indication. A response may communicate parameters that indicate its type. A data transfer response usually triggers a corresponding confirmation on a peer node.
- d) *Confirmation service.* A confirmation service is a communication from a layer to a higher or adjacent layer to confirm a request service; a confirmation is always associated with a request. A confirmation may communicate parameters that indicate the completion status of the request or that indicate other status. For data transfer requests, the confirmation may be caused by a corresponding response on a peer node.

If all four service types exist, they are related as shown by Figure 1-2.

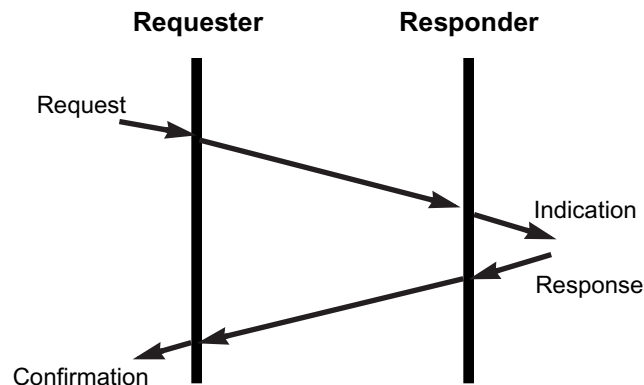


Figure 1-2—Service model

Replace 1.6 in IEEE Std 1394-1995 with the following:

1.6 Document notation

1.6.1 Mechanical notation

All mechanical drawings in this standard use millimeters as the standard unit and follow ANSI Y14.2M-1992⁸ and ANSI Y14.5M-1994 formats.

1.6.2 Signal naming

All electrical signals are shown in all uppercase characters, and active-low signals have the suffix “*”. For example, TPA and TPA* are the normal and inverted signals in a differential pair.

⁸Information on normative references can be found in 1.2.

1.6.3 Size notation

This document avoids the terms *word*, *half-word*, and *double-word*, which have widely different definitions depending on the word size of the processor. In their place, processor-independent terms [some of which have been defined by the command and status register (CSR) architecture] are used. These terms are illustrated in Table 1-1.

Table 1-1—Size notation examples

Size (in bits)	16 bit word notation	32 bit word notation	IEEE standard notation (used in this standard)
8	byte	byte	byte
16	word	half-word	doublet
32	long-word	word	quadlet
64	quad-word	double	octlet

Serial Bus uses big-endian ordering for byte addresses within a quadlet and quadlet addresses within an octlet. For 32 bit quadlet registers, byte 0 is always the most significant byte of the register. For a 64 bit quadlet-register pair, the first quadlet is always the most significant. The field on the left (most significant) is transmitted first; within a field the most significant bit (msb), i.e., the leftmost, is also transmitted first. This ordering convention is illustrated in Figure 1-3.

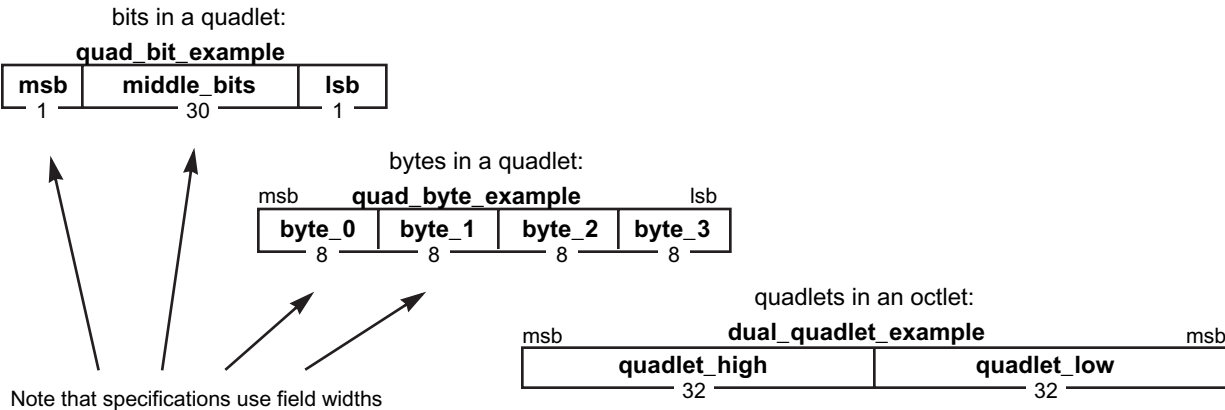


Figure 1-3—Bit and byte ordering

Although Serial Bus addresses are defined to be big-endian, their data values may also be processed by little-endian processors. To minimize the confusion between conflicting notations, the location and size of bit fields are usually specified by width, rather than their absolute positions, as is also illustrated in Figure 1-3.

When specific bit fields must be used, the CSR architecture convention of consistent big-endian numbering is used. Hence, the msb of a quadlet will be labeled “quad_bit_example[0],” the most significant byte of a quadlet (“byte_0”) will be labeled “quad_byte_example[0:7],” and the most significant quadlet in an octlet (“quadlet_high”) will be labeled “dual_quadlet_example[0:31].”

The msb shall be transmitted first for all fields and values defined by this standard, including the data values read or written to CSRs.

1.6.4 Numerical values

Decimal, hexadecimal, and binary numbers are used within this document. For clarity, the decimal numbers are generally used to represent counts, hexadecimal numbers are used to represent addresses, and binary numbers are used to describe bit patterns within binary fields.

Decimal numbers are represented in their standard 0, 1, 2, ... format. Hexadecimal numbers are represented by a string of one or more hexadecimal (0-9, A-F) digits followed by the subscript 16. Binary numbers are represented by a string of one or more binary (0,1) digits, followed by the subscript 2. Thus the decimal number “26” may also be represented as “1A₁₆” or “11010₂”. In C code examples, hexadecimal numbers have a “0x” prefix and binary numbers have a “0b” prefix, so the decimal number “26” would be represented by “0x1A” or “0b11010”.

1.6.5 Packet formats

Serial Bus packets consist of a sequence of quadlets. Packet formats are shown using the style given in Figure 1-4.

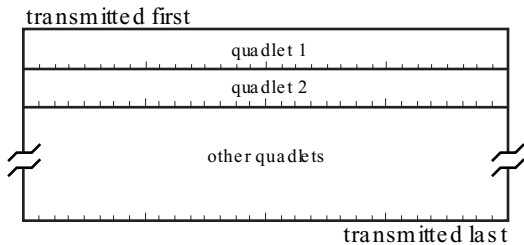


Figure 1-4—Sample packet format

Fields appear in packet formats with their correct position and widths. Field widths are also stated explicitly in field descriptions. Bits in a packet are transmitted starting with the upper leftmost bit and finishing with the bottom rightmost bit. Given the rules in 1.6.3, the most significant bit of each field defined in IEEE Std 1394b-2002 is sent first.

1.6.6 Register formats

All Serial Bus registers are documented in the style used by the CSR architecture.

1.6.7 C code notation

The conditions and actions of the state machines are formally defined by C code. Although familiar to software engineers, C code operators are not necessarily obvious to all readers. The meanings of C code operators, arithmetic, relational logical and bitwise, both unary and binary, are summarized in Table 1-2.

Table 1-2—C code operators summary

Operator	Description
+, -, * and /	Arithmetic operators for addition, subtraction, multiplication, and integer division
%	Modulus; x % y produces the remainder when x is divided by y
>, >=, < and <=	Relational operators for greater than, greater than or equal, less than, and less than or equal
== and !=	Relational operators for equal and not equal; the assignment operator, =, should not be confused with ==
++	Increment; i++ increments the value of the operand after it is used in the expression while ++i increments it before it is used in the expression
--	Decrement; post-decrement, i--, and pre-decrement, --i, are permitted.
&&	Logical AND
	Logical OR
!	Unary negation; converts a nonzero operand into 0 and a zero operand into 1
&	Bitwise AND

Table 1-2—C code operators summary (continued)

Operator	Description
	Bitwise inclusive OR
^	Bitwise exclusive OR (XOR)
<<	Left shift; $x \ll 2$ shifts the value of x left by two bit positions and fills the vacated positions with zero
>>	Right shift; vacated bit positions are filled with zero or one according to the data type of the operand, but in this amendment are always filled with zero
~	Ones complement (unary)

A common construction in C is conditional evaluation, in the form $(\text{expr}) ? \text{expr1} : \text{expr2}$. This indicates that if the logical expression expr evaluates to a nonzero value, then expr1 is evaluated. Otherwise, expr2 is evaluated. For example, $x = (q > 5) ? x + 1 : 14$ first evaluates $q > 5$. If nonzero (TRUE), x is incremented; otherwise, x is assigned the value 14.

The descriptions above are casual; if in doubt, the reader is encouraged to consult ISO/IEC 9899:1999.

The C code examples assume the data types listed in Table 1-3 are defined.

Table 1-3—Additional C data types

Data type	Description
timer	A real number, in units of seconds, that autonomously increments at a defined rate
Boolean	A single bit, where 0 encodes FALSE and 1 encodes TRUE

All C code is to be interpreted as if it could be executed instantaneously, but time may elapse when conditional expressions are evaluated as part of iterated C code. Time elapses unconditionally only when the `wait_time` function is called and may elapse when the `wait_event` function is called:

```
void wait_time(float time); // Wait for time, in seconds, to elapse
void wait_event (event); // Wait for an event to be signaled
```

The C code has been extended with a construct to allow the description of operations that execute in parallel. Two new keywords are defined:

```
fork . . . join
```

These keywords define a construct that contains a number of components that execute in parallel. The components may include statements that take time to execute (for example `wait_time(. . .)`). The construct terminates when all the components have terminated. The construct is illustrated in the following example:

```
fork
{ // first parallel component
    . . .
}
{ // second parallel component
    . . .
}
{ // final parallel component
    . . .
}
join
```

1.6.8 State machine notation

All state machines in IEEE Std 1394b-2002 use the style shown in Figure 1-5.

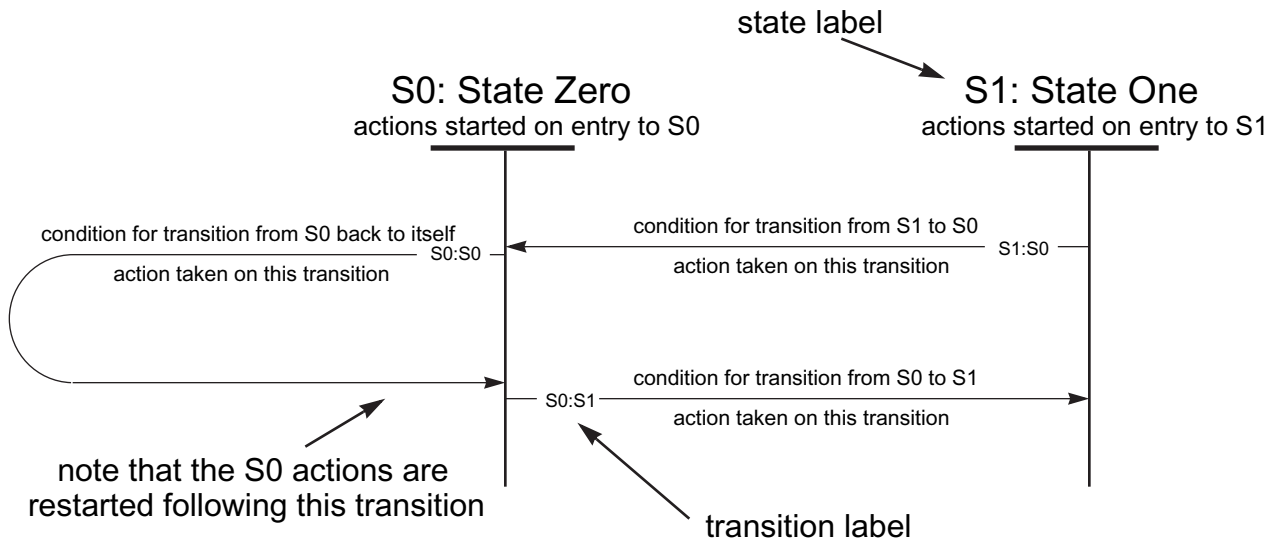


Figure 1-5—State machine example

These state machines make three assumptions:

- Time elapses only within discrete states;
- State transitions are logically instantaneous, so the only actions taken during a transition are setting flags and variables and sending signals. These actions complete before the next state is entered; and
- Every time a state is entered, the actions of that state are started. A transition that points back to the same state will repeat the actions from the beginning. All the actions started upon entry complete before any tests are made to exit the state.

1.6.9 CSR, ROM, and field notation

This standard describes a number of CSRs and fields within these registers. To distinguish register and field names from node states or descriptive text, the register name is always capitalized. Thus, the notation STATE_CLEAR.lost is used to describe the lost bit within the STATE_CLEAR register.

All CSRs are quadlets and are quadlet aligned. The address of a register (which is always a multiple of four) is specified as the byte offset from the beginning of the initial register space. When a range of register addresses is described, the ending address is the address of the last register, which is also a multiple of four. These addressing conventions are illustrated in Table 1-4.

Table 1-4—Sample CSR addressing conventions

Offset	Register name	Description
0	STATE_CLEAR	First CSR location
4-12	OTHER_REGISTERS	Next three CSR locations

This standard describes a number of configuration read-only memory (ROM) entries and fields within these entries. To distinguish ROM entry and field names from node states or descriptive text, the first character of the entry name is always capitalized. Thus, the notation Bus_Info_Block.cmc is used to describe the cmc bit within the Bus_Info_Block entry.

Entries within temporary data structures, such as packets, timers and counters, are shown in lowercase (following normal C-language conventions) and are formatted in a fixed-space typeface. Examples are `arb_timer` and `connected[ i ]`.

NOTE—Within the C code, the character formatting is not used, but the capitalization rules are followed.

1.6.10 Register specification format

This standard defines the format and function of Serial Bus-specific CSRs. Registers may be read only, write only, or both readable and writable. The same distinctions may apply to any field within a register. A CSR specification includes the format (i.e., the sizes and names of bit field locations), the initial value of the register, the value returned when the register is read, and the effect(s) when the register is written. An example register is illustrated in Figure 1-6.

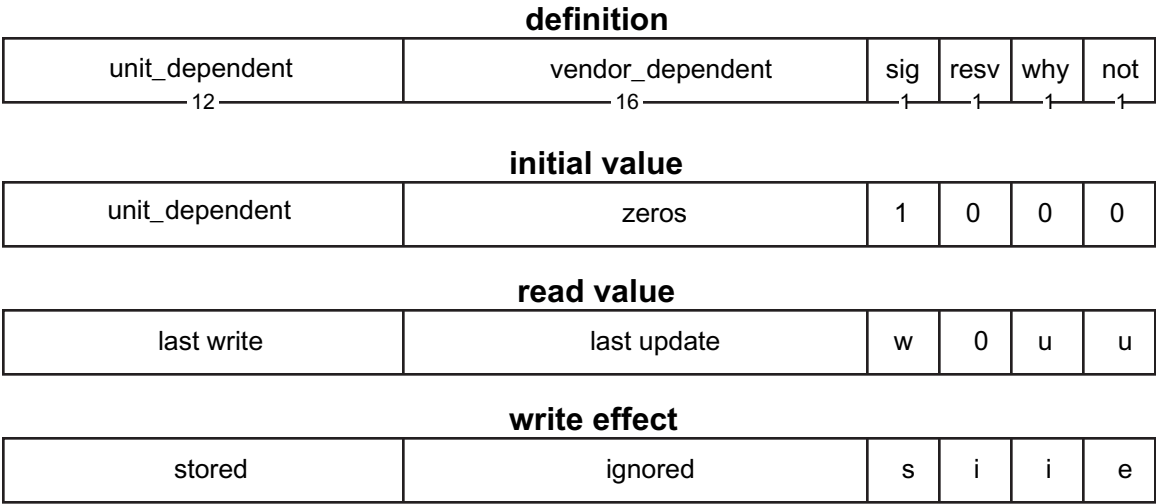


Figure 1-6—Example of CSR format specification

The register definition lists the names of register fields. These names are descriptive, but the fields are defined in the text; their function should not be inferred solely from their names. However, the register definition fields in Figure 1-6 have the meanings specified by Table 1-5.

Table 1-5—Register definition fields

Name	Abbreviation	Definition
unit dependent	unit_depend	The meaning of this field shall be defined by the unit architecture(s) of the node.
vendor dependent	vendor_depend	The meaning of this field shall be defined by the vendor of the node. Within a unit architecture, the unit-dependent fields may be defined to be vendor dependent.

A node’s CSRs shall be initialized when power is restored (power-on reset) and may be initialized when a bus reset occurs or a quadlet is written to the node’s RESET_START register (command reset). If a CSR’s bus reset or command reset values differ from its initial values, they shall be explicitly specified.

The read value fields in Figure 1-6 have the meanings specified by Table 1-6.

Table 1-6—Read value fields

Name	Abbreviation	Definition
last write	w	The value of the data field shall be the value that was previously written to the same register address.
last update	u	The value of the data field shall be the last value that was updated by node hardware.

The write-effect fields in Figure 1-6 have the meanings specified by Table 1-7.

Table 1-7—Write-effect fields

Name	Abbreviation	Definition
stored	s	The value of the written data field shall be immediately visible to reads of the same register.
ignored	i	The value of the written data field shall be ignored; it shall have no effect on the state of the node.
effect	e	The value of the written data field shall have an effect on the state of the node, but may not be immediately visible to reads of the same register.

1.6.11 Reserved registers and fields

Reserved fields within a CSR conform to the requirements of the conformance glossary in this standard (see 2.1). Within a CSR, such a field is labeled *reserved* (sometimes abbreviated as lowercase *r* or *resv*). Reserved fields behave as specified by Figure 1-7: they shall be zero, and any attempt to write to them shall be ignored.

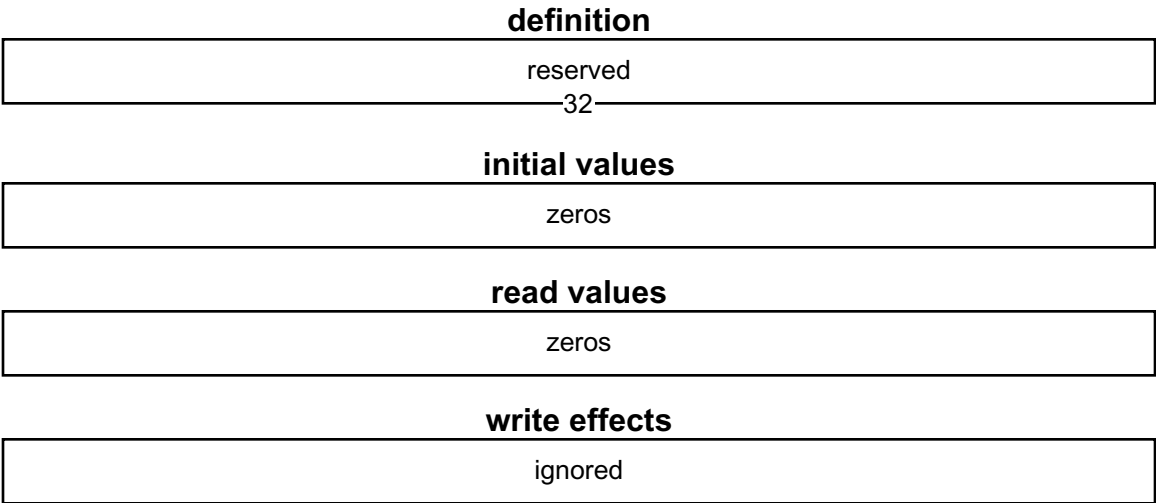


Figure 1-7—Reserved CSR field behavior

This is straight-forward as it applies to read and write requests. The same rules apply to lock requests, but the behaviors are less obvious; see IEEE Std 1394a-2000 for details.

1.6.12 Operation description priorities

The description of operations in this standard are done in three ways: state machines, C code segments, and English language. If more than one description is present, then priority shall be given first to the state machines, then the C code segments, and finally to the English text (including the state machine notes).

2. Definitions and abbreviations

Replace 2.1 in IEEE Std 1394-1995 with the following:

2.1 Conformance

Several keywords are used to differentiate between different levels of requirements and optionality, as follows:

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

2.1.2 ignored: A keyword that describes bits, bytes, quadlets, octlets, or fields whose values are not checked by the recipient.

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

2.1.4 reserved: A keyword used to describe objects—bits, bytes, quadlets, octlets, and fields—or the code values assigned to these objects where either the object or the code value is set aside for future standardization. Usage and interpretation may be specified by future extensions to this or other standards. A reserved object shall be zeroed or, upon development of a future standard, set to a value specified by such a standard. The recipient of a reserved object shall not check its value. The recipient of a defined object shall check its value and reject reserved code values.

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

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

Change the head for 2.2 in IEEE Std 1394-1995 as follows:

2.2 Technical glossary

Replace the following in 2.2 in IEEE Std 1394-1995, in alphabetical order. Replace the “x” in 2.2.x with the appropriate numbers:

2.2.x acknowledge packet: An 8 bit packet that may be transmitted in response to the receipt of a primary packet. The most and least significant nibbles are the ones complement of each other.

2.2.x arbitration: The process by which nodes compete for control of the bus. Upon completion of arbitration, the winning node is able to transmit a packet or initiate a short bus reset.

2.2.x arbitration reset gap: The period of time on a Legacy bus that indicates the start of a new asynchronous arbitration fairness interval. An arbitration reset gap is longer than a subaction gap.

2.2.x asynchronous packet: A primary packet transmitted in accordance with asynchronous arbitration rules (outside of the isochronous period).

2.2.x base rate: A data rate of 98.304 Mbit/s $\pm$ 100 ppm. In a cable environment, all Legacy-capable nodes are capable of communicating at this rate, and all B-capable nodes are capable of communicating at 4 * base rate.

2.2.x concatenated transaction: A split transaction comprising concatenated subactions.

2.2.x cycle master: The node that generates the periodic cycle start packet 8000 times a second.

2.2.x cycle start or cycle start packet: A primary packet sent by the cycle master that indicates the start of an isochronous interval.

2.2.x data bit: The smallest signaling element used by the physical layer (PHY) for transmission of packet data on the medium.

2.2.x fairness interval: A time period delimited by arbitration reset indicators. Within a fairness interval, the total number of asynchronous packets that may be transmitted by a node is limited. Each node's limit may be explicitly established by the bus manager or it may be implicit.

2.2.x initial register space: A 2048 byte portion of initial node space with a base address of FFFF F000₁₆. This address space is reserved for resources accessible immediately after a bus reset. Core registers defined by ISO/IEC 13213:1994 for CSR architecture are located within initial register space as are Serial Bus-dependent registers defined by IEEE Std 1394-1995.

2.2.x isochronous: Uniform in time (i.e., having equal duration) and recurring at regular intervals.

2.2.x isochronous resource manager: A node that implements the BUS_MANAGER_ID, BANDWIDTH_AVAILABLE, CHANNELS_AVAILABLE, and BROADCAST_CHANNEL registers (some of which permit the cooperative allocation of isochronous resources). Subsequent to each bus reset, one isochronous resource manager is selected from all nodes capable of this function.

2.2.x isochronous subaction: Within the isochronous interval, either a concatenated packet or a packet and the gap that preceded it.

2.2.x link layer or link: The Serial Bus protocol layer that provides confirmed and unconfirmed transmission or reception of primary packets.

2.2.x node: A Serial Bus device that may be addressed independently of other nodes. A minimal node consists of only a PHY without an enabled link. If the link and other layers are present and enabled they are considered part of the node.

2.2.x packet: A sequence of zero or more bits transmitted on Serial Bus and delimited by a packet start symbol and a packet end symbol.

2.2.x physical layer (PHY): The Serial Bus protocol layer that translates the logical symbols used by the link layer into electrical signals on Serial Bus media. The PHY is self-initializing. PHY arbitration guarantees that only one node at a time is sending data. The mechanical interface is defined as part of the PHY. The PHY for the backplane is different from the PHY for the cable environment.

2.2.x physical layer (PHY) packet: A 64 bit packet where the 32 most significant bits (msbs) are the ones complement of the 32 least significant bits (lsbs).

2.2.x port: The part of the physical layer (PHY) that allows connection to one other node.

2.2.x response: A primary packet (with optional data) sent in response to a request subaction.

2.2.x self-ID packet: A physical layer (PHY) packet transmitted by a cable PHY during the self-identify phase or in response to a PHY ping packet.

2.2.x unit architecture: The document that specifies the interface to, and the behaviors of, a unit implemented within a node.

Insert the following in 2.2 in IEEE Std 1394-1995, in alphabetical order. Replace the “x” in 2.2.x with the appropriate numbers:

2.2.x 3B/4B: A line code that maps 3 bit symbols to 4 bit symbols to achieve dc balance and bounded disparity. Used to create the 8B/10B code.

2.2.x 5B/6B: A line code that maps 5 bit symbols to 6 bit symbols to achieve dc balance and bounded disparity. Used to create the 8B/10B code.

2.2.x 8B/10B: A line code that maps 8 bit symbols to 10 bit symbols to achieve dc balance and bounded disparity.

2.2.x active port: A connected, enabled port that is capable of detecting all Serial Bus signal states and participating in the reset, tree identify, self-identify, and normal arbitration phases.

2.2.x arbitration request: An 8B/10B-coded symbol that is sent on a Beta port when it is not transmitting data. An arbitration request is used for bus owner/supervisor/selector (BOSS) arbitration and combines requests for both isochronous and asynchronous packet transmissions into a single symbol that, after scrambling, is represented by a single data character.

2.2.x arbitration reset: If no node has an asynchronous request for the current fairness interval, the reset that is signaled to start a new asynchronous arbitration fairness interval. In a Legacy cloud, the reset is signaled by arbitration reset gap. In a B cloud, the reset is signaled by an arbitration reset token.

2.2.x arbitration reset indication: An arbitration reset gap in a Legacy cloud or an arbitration reset token in a B cloud.

2.2.x arbitration reset token: Either an ASYNC_EVEN or an ASYNC_ODD control token, which is sent in a B cloud to indicate the start of a new asynchronous arbitration fairness interval by changing the phase.

2.2.x arbitration signaling: In a Legacy cloud, a protocol for the exchange of bidirectional, unlocked signals between nodes during arbitration. In a B cloud, the exchange of arbitration tokens.

2.2.x B bus: An operating bus in which all nodes are operating as B physical layers (PHYs).

2.2.x B cloud: A collection of B nodes and/or border nodes in which all internode connections are made through Beta ports.

2.2.x Beta mode: The mode in which a port is operating according to the specifications given in Clause 13, Clause 14, and Clause 16 and, in particular, is using the 8B/10B symbol coding and obeying the bus owner/supervisor/selector (BOSS) arbitration protocols. The speed of a port sending in Beta mode is denoted by the β suffix (e.g., S400 β).

2.2.x Beta-only port: A port that is capable of operating only as a Beta port.

2.2.x Beta port: A port that is operating in Beta mode.

2.2.x bilingual port: A port that is capable of operating both as a Beta port and as a data-strobe (DS) port. One of the modes is selected at the time that the logical connection is made; which one is selected depends on the capabilities of the peer port.

2.2.x B link: A link that is capable of operating according to the specifications given in Clause 17 or Clause 18, and, in particular, issues requests appropriate to bus owner/supervisor/selector (BOSS) arbitration.

2.2.x B node. A node whose physical layer (PHY) is operating as a B PHY.

2.2.x B-only physical layer (PHY): A PHY that is capable of only B PHY mode of operation, i.e., all its ports are Beta-only ports and its link, if any, is a B link or a PHY that is integrated into the link (PIL).

2.2.x border node: A node with both

- (i) either its link operating as a B link or at least one Beta port or both, and
- (ii) either its link operating as a Legacy link or at least one data-strobe (DS) port or both.

2.2.x B-parallel link: A mode of link operation in which the PHY-link signalling is provided to the physical layer (PHY) using a parallel interface.

2.2.x B physical layer (PHY): A mode of PHY operation in which all the logically connected ports are operating in Beta mode and the link, if any, is *not* configured to operate in Legacy PHY-link mode.

2.2.x bus identifier (ID): A 10 bit number uniquely specifying a particular bus within a system of multiple interconnected buses.

2.2.x bus owner/supervisor/selector (BOSS): In a B cloud, the node currently responsible for making arbitration decisions. A node becomes BOSS by virtue of being the last node to transmit data in a subaction (in general, the node transmitting an acknowledge to a nonbroadcast asynchronous packet or the primary packet transmitter in all other cases) or by receiving a grant. The BOSS determines the end of the fairness interval and the end of isochronous intervals and notifies the other nodes. Finally, the BOSS selects the path to grant next, thereby passing the BOSS rights and responsibilities to another node.

2.2.x bus owner/supervisor/selector (BOSS) arbitration: The arbitration scheme defined by IEEE Std 1394b-2002. The principle features of BOSS arbitration are that the node making the arbitration decision varies, that arbitration requests may be overlapped with data transmission, and that both isochronous and asynchronous requests may be pipelined for the succeeding isochronous interval or fairness interval, respectively.

2.2.x character: A 10 bit sequence of data sent on a connection that is operating in B mode.

2.2.x comma character or comma sequence: A bit sequence that defines boundaries for byte synchronization.

2.2.x configuration request: A request that conveys information pertaining to the configuration of the bus or supporting non-bus owner/supervisor/selector (non-BOSS) arbitration. A configuration request is encoded into a single symbol that, after scrambling, is represented by a data character.

2.2.x connection: Two ports that can communicate with each other and the media between those two ports.

2.2.x connection attenuation: The static loss of a connection between a transmitter and receiver. It includes the loss of the fiber, connectors, and splices.

2.2.x connection penalties: The power penalties of a connection not attributed to connection attenuation. It includes modal noise, relative intensity noise (RIN), inter-symbol interface (ISI), mode partition noise, extinction ratio, and eye opening penalties.

2.2.x connection tone: Signal used to indicate that a port is capable of operating in Beta mode. Also confirms that a connection exists between two Beta mode capable ports.

2.2.x control character: A 10 bit character used to convey control information. Control characters are distinguishable from data characters in that they are not 8B/10B characters. Each control character has zero disparity.

2.2.x data character: A character used by the 8B/10B code.

2.2.x data-strobe (DS): A signaling method using two signals in which one signal (data) always indicates the binary value of the data (0 or 1) and the other signal (strobe) changes if the data stays the same during successive bit cells. This signaling method is used in IEEE Std 1394-1995 and IEEE Std 1394a-2000.

2.2.x data-strobe (DS) mode: The form of electrical signaling and handshaking that is compatible with IEEE Std 1394-1995 and IEEE Std 1394a-2000.

2.2.x data-strobe- (DS-) only port: A port capable of operating only as a DS port.

2.2.x data-strobe (DS) port: A port that is operating according to Legacy specifications, in particular using DS electrical signaling and obeying the arbitration protocols defined therein. A DS-only port or a bilingual port can operate as a DS port.

2.2.x dc balance: The requirement that over long runs of binary symbols no net disparity shall exist.

2.2.x disabled port: A port configured to not transmit, receive, or repeat Serial Bus signals. A disabled port shall be reported as disconnected in a physical layer's (PHY's) self-ID packet(s).

2.2.x disconnected port: A port whose connection detect circuitry detects no peer physical layer (PHY) at the other end of a cable.

2.2.x disparity: The number of ones in a transmission character minus the number of zeros in a character.

2.2.x dispersion slope: The rate of change of the chromatic dispersion of a fiber with wavelength.

2.2.x extinction ratio: The ratio of the average optical power representing a logical 1 to the average optical power representing a logical 0 measured when transients have settled.

2.2.x eye diagram: Oscilloscope display of the physical layer (PHY) signal that is triggered on a bit edge and shows many different bit patterns overlaid on top of each other.

2.2.x fan-out PHY (FOP): A multiported physical layer (PHY) that is attached to a PHY that is integrated into the link (PIL) using the serial interface defined in Clause 18.

2.2.x fiber attenuation: The static loss per unit length of the fiber at a particular wavelength, usually expressed in decibels per kilometer.

2.2.x galvanic isolation: A mechanism to avoid low-frequency ground loop currents.

2.2.x hybrid bus: An operating bus that contains at least one border node.

2.2.x initial node space: The 256 terabytes of Serial Bus address space that is available to each node. Addresses within initial node space are 48 bits and are based at zero. The initial node space includes initial memory space, private space, initial register space, and initial units space. See either ISO/IEC 13213:1994 or IEEE Std 1394-1995 for more information on address spaces.

2.2.x initial units space: A portion of initial node space with a base address of FFFF F000 800₁₆. Initial units space is adjacent to and above initial register space. The command and status registers (CSRs) and other facilities defined by unit architectures are expected to lie within this space.

2.2.x isochronous interval: A period that begins after a cycle start packet is sent and ends with a subaction indication. During an isochronous interval, only isochronous subactions may occur. An isochronous interval begins, on average, every 125 μ s.

NOTE—This term appeared in IEEE Std 1394a-2000 as “isochronous period.”

2.2.x isolated node: A node without active ports; the node's ports may be disabled, disconnected, or suspended in any combination.

2.2.x jitter: Any variation in the zero-crossing time from the ideal bit pattern.

2.2.x junior border: Any border node in a B cloud except the senior border.

2.2.x K-code: An 8B/10B symbol that represents control information.

2.2.x Legacy: Characteristics or behavior of a link, node, physical layer (PHY), cable, or connector defined by IEEE Std 1394-1995 and IEEE Std 1394a-2000.

2.2.x Legacy cloud: A collection of Legacy nodes and/or border nodes in which all internode connections are made through Legacy ports.

2.2.x Legacy request: A request type defined by this specification that is originated by a border node on behalf of a Legacy link or a data-strobe (DS) port.

2.2.x link: Link layer. Also, the physical entity that implements the link layer. Also, one of the components connected to the PHY-link interface.

2.2.x logically connected port: A port whose “connected” status is true. If a port is physically connected to a peer that is not powered, then the port is not logically connected.

2.2.x low-power connection signaling: Signaling, based on the exchange of very low duty cycle tones, by which the connectivity status of a port is determined. This signaling takes place when the port is not active or is disabled.

2.2.x modal bandwidth: The bandwidth of a multimode fiber (MMF) due to dispersion caused by variations in speed of the propagating modes.

2.2.x mode partition noise: Amplitude, frequency, and phase noise in the detected optical signal due to the interaction of the modes of a multimode laser and the optical dispersion of the connection.

2.2.x near end crosstalk (NEXT): The noise induced in the receiving pair due to the signal on the transmitting pair on the same port. For example, the signal on the TPB pair can cause NEXT on the TPA pair of a port.

2.2.x nephew node: A node with one port in standby and all other ports (if any) disabled, disconnected, or suspended. The peer uncle node proxies for the nephew node during bus resets.

2.2.x node identifier (ID): A 16 bit number that uniquely differentiates a node from all other nodes within a group of interconnected buses. The 10 most significant bits (msbs) of node ID are the same for all nodes on the same bus; this number is the bus ID. The 6 least significant bits (lsbs) of node ID are unique for each node on the same bus; this number is called the physical ID. The physical ID is assigned as a consequence of bus initialization.

2.2.x null packet: A packet in which no data are transmitted.

2.2.x operating speed: Nominal speed at which a port operating in Beta mode is communicating clocked information with its peer, measured in megabits per second (before 8B/10B encoding), usually identified with the “S” notation (e.g., S100, S200, S400).

2.2.x optical power budget: The minimum optical power available to overcome the sum of attenuation plus power penalties of the optical path between the transmitter and receiver calculated as the difference between the transmitter launch power (minimum) and the receive sensitivity (minimum).

2.2.x originating port: A transmitting port on a physical layer (PHY) that has no active receiving port. The source of the transmitted packet is either the PHY’s local link or the PHY itself.

2.2.x overfilled launch (OFL): The condition that excites both radial and azimuthal modes defined in TIA/EIA 455-54B.

2.2.x packet delimiter: Sequence of control symbols that delineate the beginning and ending of a data packet.

2.2.x packet speed: The data rate of a packet (necessarily less than or equal to the operating speed of a Beta-mode connection used to transmit the packet).

2.2.x padding: Symbols inserted between data symbols in a packet to allow slower speed data to be transmitted on a higher speed connection.

2.2.x physical identifier (ID): The 6 least significant bits (lsbs) of the node ID. On a particular bus, each node’s physical ID is unique.

2.2.x physical layer (PHY) integrated link (PIL): A link that uses a modified Beta port to attach to a fan-out PHY (FOP) using the protocols defined in Clause 18.

- 2.2.x physical media dependent (PMD) interface:** The part of an interface that is specific to single kind of interconnect.
- 2.2.x ping:** A term used to describe the transmission of a physical layer (PHY) packet to a particular node in order to time the response packet(s) provoked.
- 2.2.x point-to-point (P2P) packet:** Special packet type that is sent on the PIL-FOP interface. This packet is used to carry data that cannot be sent as part of a normal Serial Bus packets.
- 2.2.x port state machine:** The logic that controls the functioning of a port.
- 2.2.x proxy_root:** In a B-only bus, the root. In a hybrid bus where the root is in a B cloud, the senior border in the B cloud containing the root. If the root is a Legacy node, then no proxy_root exists.
- 2.2.x radial overfilled launch (OFL):** A launch condition created when a multimode optical fiber (MMF) is illuminated by the coherent optical output of a source operating in its lowest order transverse mode in a manner that excites predominantly the radial modes of the MMF.
- 2.2.x random jitter:** Jitter that comes from random sources. Characterized by Gaussian statistics and unbounded variation according to the Gaussian distribution function.
- 2.2.x receiver eye opening:** The interval in time within a bit period where the sampled data value will have a probability of error less than the specified bit error ratio (BER).
- 2.2.x register:** A term used to describe addresses that may be read or written by Serial Bus transactions. In the context of IEEE Std 1394b-2002, the use of the term *register* does not imply a specific hardware implementation. For example, for split transactions that permit sufficient time between the request and response subactions, the behavior of the register may be emulated by a processor within the module.
- 2.2.x relative intensity noise (RIN):** The ratio of the variance in the optical power to the average optical power.
- 2.2.x repeating port:** A transmitting port on a physical layer (PHY) that is repeating a packet from the PHY's receiving port.
- 2.2.x request type:** An 8B/10B data character that connotes either an arbitration or configuration request.
- 2.2.x restore:** The process of causing a connection in standby to return to the active state.
- 2.2.x resume signal:** A signal requiring the port to resume normal operations.
- 2.2.x resuming port:** A previously suspended port that has observed signaling other than the connection tone or has been instructed to resume. In either case, the resuming port engages in a protocol with its connected peer physical layer (PHY) to reestablish normal operations and become active.
- 2.2.x rms spectral width:** The optical wavelength range as measured by ANSI/TIA/EIA-455-127-91.
- 2.2.x run length:** The length of a sequence of bits that have the same value, e.g., ones or zeros.
- 2.2.x running digital sum:** The number of ones in a character stream minus the number of zeros in a character stream.
- 2.2.x running disparity:** An estimate of the running digital sum at the end of a character subblock, based upon the most recently transmitted (or received) character subblock. When initialized with the same value, the running disparity and running digital sum will be equal at the end of any character subblock. Errors in a received character stream may result in the running disparity being not equal to the running digital sum.

2.2.x scrambler: A device that chooses transmitted signals from the set of all possible symbols by using both the desired symbol and a pseudo-randomly generated offset pointer. Used to average out the spectral content of the transmitted signal to avoid strong spectral lines.

2.2.x senior border: A unique border node in a B cloud. The senior border is the last node to originate a self-ID packet without a speed code into the cloud (i.e., repeated from a data-strobe (DS) port or generated because of a Legacy link) and is responsible for ensuring that certain Legacy gap timings are observed.

2.2.x standby: A low-power state of a Beta connection in which only low-power connection signaling takes place. No bus reset is generated as a result of a port entering or leaving the Standby state.

2.2.x standby initiator: An active port that transmits the STANDBY configuration request and engages in a protocol with its connected peer physical layer (PHY) to place the connection into the Standby state.

2.2.x subaction gap: In a Legacy cloud, the period of idle bus that precedes arbitration for an asynchronous subaction.

2.2.x subaction indication: A subaction gap or, in a B cloud, an ASYNC_EVEN/ODD control token of the current phase.

2.2.x suspend: To go into a low-power mode of operation while maintaining low-power connection signaling. When a port enters or leaves the Suspended state, a bus reset is generated.

2.2.x suspended node: An isolated node with at least one port that is suspended.

2.2.x suspended port: A connected port not operational for normal Serial Bus arbitration, but otherwise capable of detecting either a physical cable disconnection or a resume signal.

2.2.x suspend initiator: An active port that transmits the SUSPEND configuration request or the TX_SUSPEND signal and engages in a protocol with its connected peer physical layer (PHY) to place the connection in the Suspended state.

2.2.x suspend target: An active port that receives the SUSPEND configuration request or observes the RX_SUSPEND signal. A suspend target requests that all of the other active ports on the physical layer (PHY) become suspend initiators while the suspend target port engages in a protocol with its connected peer PHY to suspend the connection.

2.2.x synchronization: The process of aligning the receiver's circuits to properly detect received bits and to properly detect symbol boundaries.

2.2.x transaction layer: The Serial Bus protocol layer that defines a request-response protocol for read, write, and lock operations.

2.2.x transmitting port: Any port transmitting clocked data or an arbitration state. A transmitting port is further characterized as either originating or repeating.

2.2.x uncle node: A node that proxies for a peer nephew node while the connection between the two is in Standby state.

2.2.x unit: A component of a Serial Bus node that provides processing, memory, input and output (I/O), or some other functionality. Once the node is initialized, the unit provides a command and status register (CSR) interface. A node may have multiple units, which normally operate independently of each other.

2.2.x unit interval: The nominal amount of time 1 bit takes to transmit.

2.2.x zero dispersion wavelength: The wavelength where the chromatic dispersion of a fiber is at its minimum.

Add the following new subclause at the end of Clause 2 in IEEE Std 1394-1995:

2.3 Acronyms and abbreviations

A bit	asynchronous phase bit
ACK	acknowledge or acknowledgment
AWG	American Wire Gauge
BER	bit error ratio
CAT-5	Category 5
CPR	coupled power ratio
CRC	cyclic redundancy code
DMD	differential-mode delay
FIFO	first in, first out buffer
FOP	fan-out PHY
FWHM	full width at half maximum
G bit	generation number bit
GOF	glass optical fiber
HPCF	hard polymer clad fiber
HR	holding register
LED	light-emitting diode
LPS	link power status
LTD	loop test data
LTP	loop test packet
LTS	loop test symbol
M bit	mode bit
MMF	multimode fiber
NA	numerical aperture
N bit	reset notify bit
NEXT	near end crosstalk
NRZ	nonreturn to zero
OFL	overfilled launch
PCB	printed circuit board
PIL	PHY that is integrated into a link
pk-pk	peak-to-peak
PLL	phase locked loop
PMD	physical media dependent
POF	plastic optical fiber
POR	power-on reset
R bit	root bit
RIN	relative intensity noise
SMF	single-mode fiber
STP	shielded twisted pair
T bit	gap count reset disable bit
TDR	time domain reflectometry
UTP	unshielded twisted pair
VCSEL	vertical cavity surface emitting laser

3. Summary description

Insert the following after 3.9, which was added to IEEE Std 1394-1995 by IEEE Std 1394a-2000:

3.10 New features of IEEE Std 1394b-2002

IEEE Std 1394b-2002 is an amendment to IEEE Std 1394-1995 as amended by IEEE Std 1394a-2000. IEEE Std 1394b-2002 defines mechanisms that increase the data rate and the transmission distance for IEEE 1394 buses.

3.10.1 The relationship to IEEE Std 1394a-2000

IEEE Std 1394a-2000 improves many areas of IEEE Std 1394-1995. Of particular note is the improvement in bus efficiency. IEEE Std 1394a-2000 provides arbitration acceleration, improves the speed of bus resets, and adds suspend and resume power management capabilities to the bus. IEEE Std 1394b-2002 is fully interoperable with IEEE Std 1394a-2000 and IEEE Std 1394-1995 at the signaling level of their 6-pin and 4-pin connectors. IEEE Std 1394b-2002 supports the DS form of signaling used in IEEE Std 1394a-2000 and IEEE Std 1394-1995 while adding a new form of signaling capable of much higher data rates between Beta ports. IEEE Std 1394b-2002 also supports new media, such as Category 5 (CAT-5) unshielded twisted pairs (UTPs), glass optical fiber (GOF), and plastic optical fiber (POF).

3.10.2 Faster and further

IEEE Std 1394b-2002 supports the regular DS signaling modes and speeds of IEEE Std 1394-1995. Additionally, IEEE Std 1394b-2002 extends bus speeds to S800 and S1600 and has architectural support for S3200—although insufficient technical data were available when IEEE Std 1394b-2002 was prepared to specify the signaling parameters for S3200. Silicon devices that operate in both the DS signaling mode and the Beta signaling mode are called *border nodes* because they are at the border of the two different signaling bus segments.

IEEE Std 1394-1995 recommends a maximum cable length of 4.5 m. Many applications find this length too short a cable run for their needs. IEEE Std 1394b-2002 supports optical cable lengths of 50 m for POF and 100 m for GOF. Additionally, IEEE Std 1394b-2002 supports S100 operation over lengths of CAT-5 up to 100 m. This extra run length permits new applications for IEEE 1394 buses, such as home networking.

3.10.3 Nomenclature

Figure 3-33 illustrates the PHY architectural framework and architectural elements possible within a Beta-capable PHY. This figure is referenced by many other subclauses in IEEE Std 1394b-2002; when reproduced in other subclauses, the context of the subclause within the architectural framework is shown by bold lines around functional components.

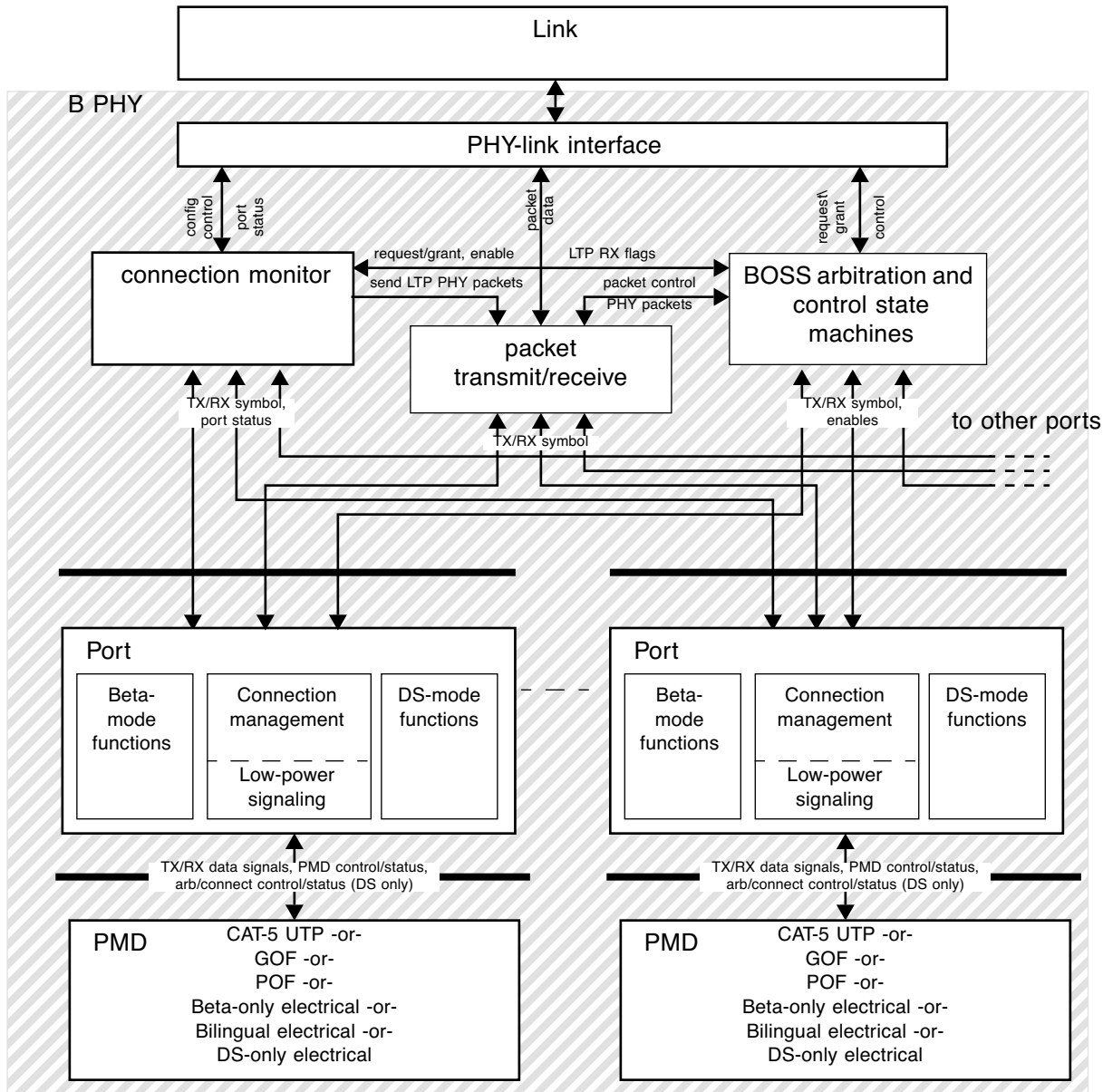


Figure 3-33 — Master architecture

3.10.4 Media—common properties

All media that are used in Beta mode use the same encoding scheme (i.e., a modified version of 8B/10B), which is a form of nonreturn to zero (NRZ) binary signaling. Two parallel physical connections (i.e., two twisted pairs or two optical fibers) continuously send data symbols in opposite directions for full-duplex operation. Symbols are scrambled before

encoding to minimize emissions. The 8B/10B encoding enforces dc balance and limits the run length of the transmitted bits that permits ac coupling and excellent control of signal integrity. All media forms are specified with the goal of having the bit error ratio (BER) less than 10^{-12} .

All media types have the same connection management. This allows the use of the same PHY for different media (with the use of a physical media dependent (PMD) module, e.g., a vertical cavity surface emitting laser (VCSEL) and photodiode for optical fiber). Thus, connecting an optical transceiver to a standard PHY is a simple matter. The dc balance feature of the symbol encoding permits the use of capacitors on the signal leads to achieve galvanic isolation.

3.10.4.1 Short-haul shielded twisted pair (STP) cabling

IEEE Std 1394b-2002 defines a new connector and its associated cable providing a roadmap to S3200. A device that will run only in Beta mode shall use the new Beta connector. DS-mode connection is achieved by using a cable with a Legacy plug on one side and a bilingual plug on the other. A bilingual or Beta-only PHY port shall not be connected to either the 4- or 6-circuit sockets defined by IEEE Std 1394-1995 or IEEE Std 1394a-2000.

The Beta connector is a small connector only slightly larger than the 4-circuit IEEE Std 1394a-2000 connector. The Beta connector includes power pins and can be keyed to indicate bilingual capability. The keying is arranged so that a bilingual plug will not mate with a Beta-only socket.

The maximum cable length remains 4.5 m, as recommended by IEEE Std 1394-1995. Longer copper lengths are possible with special cables (e.g., thicker gauge wires), but are not specified in IEEE Std 1394b-2002 and require special care in complying with all the requirements of emissions and electrical interoperation.

3.10.4.2 CAT-5 UTP cable (see Chapter 7 of ISO/IEC 11801:2000)

CAT-5 UTP cable is low cost, easily installed, and readily available. CAT-5 is well known because of its use in the Ethernet standard.

IEEE Std 1394b-2002 uses CAT-5 in a way that is compatible with typical 100BASE-T2 Ethernet cable plant. Data are transmitted out on pin 1 and pin 2 and received on pin 7 and pin 8. This wiring matrix avoids erroneous connection to Ethernet equipment. Inputs and outputs to CAT-5 are transformer isolated. IEEE Std 1394b-2002 supports autocrossover for UTP. The electrical specifications are as other standards using CAT-5, e.g., 1 V peak-to-peak (pk-pk) binary signaling. By requiring adaptive equalization in the receiver, IEEE Std 1394b-2002 achieves a maximum CAT-5 operating length of 100 m at S100. Higher speeds are not defined by IEEE Std 1394b-2002.

3.10.4.3 Plastic optic fiber (POF) and hard polymer clad fiber (HPCF)

POF consists of 1000 μm step index multimode plastic optic fiber. The optical power budget in IEEE Std 1394b-2002 permits operation up to 50 m at speeds of up to S200.

Hard polymer clad fiber (HPCF) consists of 225 μm graded index multimode hard polymer clad fiber and is suitable for operation up to 100 m at speeds up to S200.

The same transceiver specification is used for both kinds of plastic fiber.

POF and HPCF are low cost and easy to install. Fiber alleviates emissions and interference problems. A simple PN style connector is specified.

3.10.4.4 Glass fiber

The GOF specification leverages VCSEL technology. IEEE Std 1394b-2002 follows the Fibre Channel and gigabit Ethernet specifications closely. The GOF specification uses 50 μm multimode fiber (MMF), and the optical power budget gives a media roadmap of up to S3200 at 100 m.

The GOF connector is the LC connector as defined in the FOCIS 10 addendum of the TIA/EIA-604-93.

3.10.4.5 Media summary for Beta-mode operation

Table 3-1 summarizes the usage of various media that are defined in this standard. Entries with an “X” indicate the allowed media type and speed combinations

Table 3-1 — Beta-mode media summary

Media	Reach (m)	S100	S200	S400	S800	S1600
CAT-5 UTP	100	X				
POF	50	X	X			
HPCF	100	X	X			
MMF	100			X	X	X
STP	4.5			X	X	X

3.10.5 Arbitration improvements

IEEE Std 1394b-2002 makes use of the continuous full-duplex nature of Beta-mode signaling to reduce the inefficiencies of arbitration idle gaps that are integral to the arbitration methods specified by IEEE Std 1394-1995. The gap time is a function of bus topology, dependent upon PHY repeater delays, cable lengths, and hop counts for a particular bus, and is invariant for all transmission speeds. Thus, as transmission speed increases, the idle time occupied by the gap accounts for a larger percentage of total bus occupancy time for a packet. These inefficiencies are addressed somewhat by the enhanced arbitration mechanisms in IEEE Std 1394a-2000, which also scale well into the speed range defined by IEEE Std 1394b-2002.

IEEE Std 1394b-2002 further reduces these inefficiencies using a new arbitration mechanism that allows arbitration request signaling to be pipelined and overlapped with data transmission on the full-duplex bus. This new mechanism eliminates idle gaps in a B bus as long as data exist to transmit.

3.10.5.1 Legacy arbitration

In Legacy IEEE 1394, arbitration for the bus involved the following steps:

- Wait for the bus to go idle for some minimum gap time,
- Signal to arbitrate for the bus,
- Upon receiving a grant signal, send the data packet, and
- Wait for the responding node to return an acknowledge packet (for asynchronous subactions).

Idle time occurs between each of these steps.

For asynchronous arbitration as defined by IEEE Std 1394-1995, the subaction gap time is greater than the round-trip delay time for the most distance pair of nodes on the bus. Waiting for a subaction gap time after passage of the last packet before the start of new arbitration ensures that new arbitration will not interfere with an acknowledge packet. This method of bus control is acceptable for large packets where bus efficiencies can remain above 80% on a large bus. For small packets (e.g., 64 byte data payload), the overhead time can reduce the bus efficiency to around 20% on a large bus running at S100 speeds. Increasing the bus speed to S400 reduces the large data packet efficiency to perhaps 50% and the small data packet efficiency to perhaps 10%. Thus, increasing the bus speed does not yield a proportionate increase in throughput.

To increase bus efficiency, IEEE Std 1394a-2000 defines several arbitration enhancements and acceleration mechanisms:

- a) ACK-accelerated arbitration. This enhancement allows a node to begin arbitration for the bus at any time following detection of an acknowledge packet (indicating the completion of a subaction) without first waiting for a subaction idle gap to occur.

- b) Fly-by concatenation. This enhancement allows a node to concatenate a packet onto the end of an acknowledge packet or isochronous packet received on a child port without first arbitrating for the bus.
- c) Multispeed concatenated packets. This enhancement allows a node to transmit concatenated packets of differing speeds, within certain criteria, without arbitrating again each time transmission speed is changed.
- d) Priority arbitration for PHY and response packets. This enhancement permits transmission of PHY packets or response packets using priority arbitration instead of fair arbitration and thereby eliminates the additional idle time required for an arbitration reset gap.
- e) Priority budget (fairness optimization). This enhancement permits the transmission of multiple asynchronous request packets by the same node within a single fairness interval as determined by a priority budget set implicitly or explicitly by the bus manager and thereby eliminates the additional idle time required for an arbitration reset gap.

Although these arbitration enhancements and accelerations can result in increased bus efficiency, they do not entirely eliminate idle gaps and their resultant inefficiency; e.g., a subaction gap is still required to indicate the end of the isochronous transmission interval, a subaction gap still occurs after an asynchronous broadcast packet or PHY packet that does not require an acknowledge, and arbitration reset gaps are still required to delimit fairness intervals. Furthermore, as data transmission rates become higher and/or interconnect distances become longer, the time required for arbitration request and grant signaling becomes greater relative to the packet data transmission time and again reduces bus efficiency. When transmission rates extend into the gigabit-per-second range and interconnect distances approach 100 m, as defined in IEEE Std 1394b-2002, the bus efficiency attainable with the Legacy arbitration mechanisms becomes unacceptably low.

3.10.5.2 New arbitration method—bus owner/supervisor/selector (BOSS)

IEEE Std 1394b-2002 retains the arbitration enhancements and accelerations of IEEE Std 1394a-2000, but further improves bus efficiency by taking advantage of the full-duplex nature of Beta-mode signaling to implement a new arbitration method referred to as bus owner/supervisor/selector (BOSS). In BOSS operation, arbitration request signaling is overlapped with data transmission on the full-duplex bus, and both isochronous and asynchronous requests may be pipelined for servicing in the succeeding isochronous interval or fairness interval, respectively.

The BOSS arbitration scheme can work either in a bus where all nodes are compliant to this specification (i.e., a B bus) or in a bus with any mixture and configuration of nodes compliant to IEEE Std 1394b-2002 and Legacy nodes (i.e., a hybrid bus). This subclause introduces the concepts of BOSS operation as it works in a B bus and then describes the extensions to BOSS arbitration to support hybrid buses (see 3.10.5.2.4).

Figure 3-34 illustrates the general concept of BOSS operation and shows how arbitration signaling is overlapped with data transmission on the full-duplex bus.

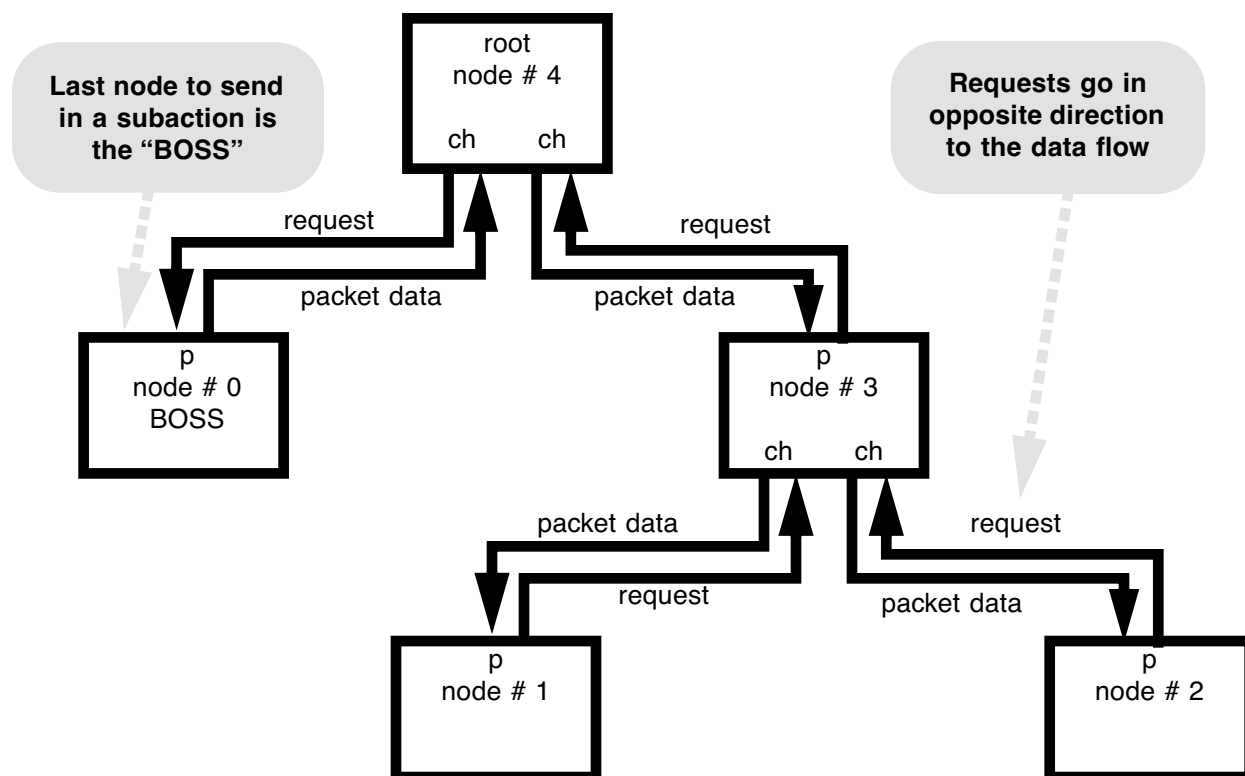


Figure 3-34—BOSS pipelined, overlapping arbitration

In general, a PHY transmits arbitration requests on any active port that is not transmitting packet data. The PHY that is the last to transmit packet data in a subaction (i.e., the PHY transmitting an acknowledge to a directed nonbroadcast asynchronous packet, or the primary packet originator in all other cases) becomes the “BOSS” and is responsible for taking the next arbitration decision. During packet transmission, the originator is the only PHY in the bus that can be receiving arbitration requests on every active port and is, therefore, in the best position to observe arbitration requests and to decide which node gets the bus next.

3.10.5.2.1 Establishing the BOSS

A PHY that originates packet transmission becomes BOSS immediately. Thus, transmission of an acknowledge or PHY response packet establishes a PHY as BOSS.

A PHY that receives an explicit or implicit grant becomes BOSS. An explicit grant occurs when a PHY receives GRANT or GRANT_ISOCH symbols as the terminator of a received packet or in response to an arbitration request. An implicit grant occurs when a receiving PHY can independently determine that the subaction has concluded (e.g., the bus is in the isochronous interval or the last packet was an acknowledge packet in the asynchronous interval).

The root in a B bus assumes the role of BOSS after any extended period of inactivity (i.e., it assumes that an acknowledge packet was missing or some other error occurred).

A PHY ceases to be BOSS when it receives or when it issues an explicit or implicit grant.

3.10.5.2.2 Tokens and bus intervals

In contrast to Legacy arbitration, where idle gaps are used to mark the end of the self-identify phase of bus initialization, to terminate the isochronous interval, to indicate the end of a subaction in the asynchronous interval, and to indicate a fairness interval boundary, tokens are used in BOSS operation for these purposes. A token is a specially designated Beta control symbol that is transmitted by the current BOSS for a specified minimum time to ensure reception throughout the bus. Tokens can be transmitted by the BOSS when the bus is otherwise idle or being held by transmission of DATA_NULL symbols (i.e., the token is embedded within the DATA_NULL symbol stream).

As in Legacy operation, packet transmission in BOSS operation is divided into isochronous and asynchronous intervals. In a B bus, the isochronous interval begins when a CYCLE_START_EVEN/ODD token and cycle start packet are transmitted by the cycle master. The isochronous interval ends when the current BOSS has no in-phase isochronous requests pending either from its link or at any active Beta port, whereupon it transmits an ASYNC_EVEN/ODD token to begin the asynchronous interval. Additionally, an ASYNC_EVEN token is transmitted by the root in a B bus to indicate the end of the self-identify phase (i.e., the last phase of bus initialization) and the start of normal arbitration.

The asynchronous interval is active when the bus is not in the isochronous interval. As in Legacy operation, the asynchronous interval is further divided into fairness intervals. In a B bus, the fairness interval boundary is marked by the current BOSS's transmitting an ASYNC_EVEN/ODD token as appropriate for the new phase when it has no in-phase asynchronous requests pending either from its attached link or at any active Beta port. It is possible for the BOSS to terminate both an isochronous interval and a fairness interval using a single transmission of an ASYNC_EVEN/ODD token.

To increase robustness against dropped tokens, the root in a B bus transmits ASYNC_EVEN/ODD tokens continuously when the bus is idle.

3.10.5.2.3 Arbitration requests, pipelining, and phasing

In BOSS mode, arbitration is performed with 8B/10B Beta symbols. Unlike Legacy arbitration, which has only one line state to signal an arbitration request, these arbitration request symbols incorporate both an asynchronous and isochronous request code and allow multiple request types and priorities to be implemented.

The availability of multiple request types and priorities, in turn, facilitates arbitration pipelining, which is a mechanism whereby requests meant to be serviced in the next isochronous interval or asynchronous fairness interval are transmitted and available to the BOSS far in advance of the BOSS's having to make an arbitration decision on them. To fully facilitate pipelining, the concept of *arbitration phase* is introduced, where the arbitration phase alternates between even and odd. Both the isochronous and asynchronous intervals have phase, each of which are independent of one another. The isochronous phase advances at the end of the isochronous period (as indicated by the transmission of an ASYNC_EVEN/ODD token), and the asynchronous phase advances at each fairness interval boundary (as indicated by the transmission of an ASYNC_EVEN/ODD token corresponding to the new phase). Following bus reset, both the isochronous and asynchronous phases are defined to be even.

An arbitration request may be designated for the current phase (either even or odd) or the next or opposite phase (i.e., pipelined). Beta PHYs constantly forward the highest priority asynchronous and isochronous requests (combined into a single request symbol) received from either its attached link or any active Beta ports towards the current BOSS. The priority of arbitration requests is dependent upon the current bus interval and phase.

Following the transmission of a packet at the end of a subaction, the current BOSS immediately grants the highest priority pending request for the current interval and phase, if any. However, if at this time the BOSS is receiving no valid in-phase requests, it transmits one or more tokens to advance the bus interval and/or phase:

- a) If the bus is in the isochronous interval at the end of packet transmission and no in-phase isochronous requests are being received, the BOSS transmits an ASYNC_EVEN/ODD token to terminate the isochronous interval and advance to the asynchronous interval (but without changing the current asynchronous phase).

- b) If the bus is in the asynchronous interval at the end of packet transmission at the end of a subaction, or has entered the asynchronous interval by item a, and no in-phase asynchronous requests are being received, the BOSS transmits an ASYNC_EVEN/ODD token to advance the asynchronous phase and begin a new fairness interval.
- c) If no in-phase requests are pending for the new asynchronous phase, the BOSS grants control of the bus to its senior (i.e., the peer PHY closest to the root) so that the root eventually assumes control in an idle B bus. Once an ASYNC_EVEN/ODD token has been issued to change the phase, another token to change the phase again will not be issued until after some asynchronous packet transmission has occurred in that phase.

In the asynchronous interval, a node can make a request for the current phase only if it has any priority budget remaining for the current fairness interval, and nodes can make requests for the next asynchronous phase when they have no budget remaining. Thus, this mechanism allows fairness to be implemented in a B bus without the use of idle gaps; the bus phase is advanced only when all nodes that have data to transmit in the current fairness interval have done so.

3.10.5.2.4 Hybrid bus operation

The discussion of BOSS arbitration in 3.10.5.2.3 is specifically concerned with operation in a B bus (consisting entirely of B nodes). In a hybrid bus, this operation is modified to maintain interoperability and compatibility with Legacy devices. In particular, idle gaps must be maintained as required for the Legacy arbitration mechanisms.

Each B cloud in a hybrid bus contains only one senior border PHY, which is the last node to originate a self-ID packet without a speed code into the cloud (i.e., repeated from a DS port or generated because of a Legacy link). The senior border PHY in the B cloud that contains the root is referred to as the proxy_root and acts as the default arbiter when no other PHY is acting as BOSS. (It is possible that no proxy_root exists, as the root may be a Legacy PHY and, therefore, not part of any B cloud.) The parent port of a senior border (other than the proxy_root) is necessarily a DS port.

The senior border in each B cloud is responsible for ensuring that the Legacy gap timings are observed and is the only PHY within the cloud allowed to issue ASYNC_EVEN/ODD tokens. To synchronize Legacy gap indications with BOSS token indications, the senior border issues an ASYNC_EVEN/ODD token of the current phase when it times a subaction idle gap and issues an ASYNC_EVEN/ODD token of the new phase when it times an arbitration reset idle gap. It continues to issue the tokens for the current phase so long as the bus is idle.

In response to Beta arbitration requests, the senior border begins arbitration for the bus, but does so in accordance with the Legacy rules for initiating arbitration to ensure consistent buswide detection of gaps. A grant can be issued (if proxy_root), or arbitration on the DS parent port can begin

- a) Up to arb_delay following the issuance of an ASYNC_EVEN/ODD token indicating a subaction gap, provided the previous packet represented the end of a subaction (ACK-accelerated arbitration),
- b) At arb_delay following the issuance of an ASYNC_EVEN/ODD token indicating a subaction gap,
- c) Anytime after the arb_delay following the issuance of an ASYNC_EVEN/ODD token indicating a new fairness interval, or
- d) Within ARB_RESPONSE_DELAY after receiving a LEGACY_REQUEST or ISOCH_CURRENT request.

When the senior border wins arbitration, it begins BOSS arbitration and Beta traffic can begin. Once Beta traffic begins in a B cloud, BOSS arbitration persists until the end of a subaction has been reached and no more in-phase requests are pending from any of the nodes in the cloud.

During this period of Beta traffic, Legacy devices shall be prevented from detecting any idle gaps. Legacy packets are repeated by a border PHY directly to any DS ports as normal (with DATA_PREFIX replacing the payload if the packet speed is greater than the peer's capability). For Beta packets, a border begins generating DATA_PREFIX on its DS ports. Because the border node meets minimum DATA_PREFIX/DATA_END timings and cannot predict when the next Legacy packet will arrive, it cannot safely release DATA_PREFIX until such a packet arrives. Instead, it must hold DATA_PREFIX on the DS ports until a Legacy packet arrives. The Legacy packet is concatenated onto the DATA_PREFIX and is received normally by Legacy devices. To ensure that border PHYs do not remain in a mode that causes them to send

DATA_PREFIX indefinitely on their DS ports, the BOSS in a B cloud is required to issue a Legacy null packet when the end of a subaction has been reached, no more in-phase requests are pending, and the last packet sent was not a Legacy packet.

At the end of Beta traffic, the current BOSS does not transmit any tokens, but instead simply grants control of the bus to its senior. Control is eventually transferred to the senior border, which begins timing the idle gap.

In a hybrid bus, CYCLE_START tokens can exist only in the B cloud that contains a cycle master operating in B mode. This B cloud, if one exists, is referred to as the senior B cloud. (It is possible that no senior B cloud exists, as the cycle master may not be in a B cloud.) Any B cloud that is not the senior B cloud is referred to as a junior B cloud. Tokens cannot be transmitted across the DS connections between B clouds, and reproducing the CYCLE_START tokens in the junior B clouds is problematic.

Therefore, a PHY in a junior B cloud becomes aware of the start of the isochronous interval when it receives a PH_ISOCH_PHASE_EVEN/ODD notification from an attached isochronous-capable link. This notification also informs the PHY of the isochronous phase. Upon receiving this notification, the PHY begins sending ISOCH_CURRENT arbitration requests if it has an in-phase isochronous request pending from its link. The ISOCH_CURRENT request is the highest priority isochronous request and is eventually received by the senior border; ISOCH_EVEN/ODD requests are never generated within a junior B cloud. The senior border responds to an ISOCH_CURRENT request by immediately arbitrating for the bus.

3.10.5.2.5 Legacy request phasing

In Legacy IEEE 1394, a node arbitrates for control of the bus by sending a TX_REQUEST signal on its parent port and sending TX_DATA_PREFIX on its child ports to deny them access to the bus. In response to this arbitration request, it receives either a RX_GRANT or RX_DATA_PREFIX signal on its parent port. If RX_GRANT is received, control of the bus has been won, and the node can begin packet transmission. If RX_DATA_PREFIX is received, arbitration has been lost, and another node has been granted control of the bus to transmit a packet. The node must withdraw the TX_REQUEST on its parent port and prepare to receive the impending packet. This signaling handshake must occur before the start of clocked packet data to avoid signal contention and interference with the packet data on the twisted pair. In Legacy IEEE 1394, the various packet timing constants (e.g., MIN_DATA_PREFIX, ARB_RESPONSE_DELAY) were defined so that this signaling handshake would complete in time, but assumed a maximum round-trip delay of about 45 ns on the cable connection (corresponding to 4.5 m of copper twisted pair).

A border node converts a received RX_REQUEST signal on a DS port to Beta LEGACY_REQUEST symbols, and arbitration proceeds in a fashion similar to Legacy IEEE 1394. Junior ports are denied by sending DATA_NULL on them, and LEGACY_REQUEST is sent on the senior port. If arbitration is won, a GRANT or GRANT_ISOCH is returned. If arbitration is lost, a DATA_PREFIX or DATA_NULL is returned.

IEEE Std 1394b-2002 allows Beta-mode connections of up to 100 m on CAT-5 UTP, HPCF, or MMF media, with corresponding round-trip cable delays of well over 1 μ s. Such long cable delays present a problem for Legacy arbitration in that the signaling handshake may not complete in time. If a node is sending a LEGACY_REQUEST on its senior port and arbitration is lost, the LEGACY_REQUEST at the senior node will not be withdrawn before packet transmission begins. Because of the full-duplex nature of a Beta-mode connection, no danger of signal contention exists as on a DS-mode connection, but stale requests are possible.

A stale request can occur when a junior node begins sending LEGACY_REQUEST at about the same time that its senior node begins transmitting a packet. If the packet is short and the cable long, the senior can complete transmission of the entire packet before receiving the first indication of LEGACY_REQUEST from the junior, which will be withdrawn as soon as the junior receives the packet. Without a mechanism for recognizing this request as stale, the senior would otherwise act on the LEGACY_REQUEST, perhaps interfering with an impending ACK.

To resolve the problem of a stale LEGACY_REQUEST, the concept of *Legacy request phase* is introduced. Each LEGACY_REQUEST symbol incorporates a 2 bit field containing a Legacy request phase number, which ranges from 0 to 3. A received LEGACY_REQUEST is considered valid only if its Legacy request phase number is equal to the node's

own stored value for Legacy request phase. Otherwise, it is considered a stale request. (The Legacy request phase does not affect requests received at a DS port; a RX_REQUEST signal received on a DS port is always considered valid and is immediately acted upon.)

When a node originates a Legacy packet into a Beta cloud (either as original transmitter of the packet or when repeating a packet into a Beta cloud from a DS port) and the packet is terminated with DATA_END or with GRANT or GRANT_ISOCH on the senior port, it increments its Legacy request phase number. The new Legacy request phase number is then communicated to all other nodes in the local Beta cloud via a LEGACY_PHASE token that completes the Legacy packet's ending symbol sequence. Receiving nodes update their own Legacy request phase number with the new value contained in the LEGACY_PHASE token. If the LEGACY_PHASE token is corrupted or otherwise not received when it is expected, a receiving node increments its own Legacy request phase number and uses this value when repeating the packet. Upon bus reset, the Legacy request phase is set to 0.

3.10.6 PHY-link interface

This specification contains two new definitions for the PHY-link interface. The first of these interface definitions represents an evolutionary change from the Legacy PHY-link interface specification. The second interface is an adaptation of the high-speed, Beta-mode serial interface that offers greater flexibility in system implementation and allows higher speeds and longer distances between the PHY and the link.

3.10.6.1 Evolutionary PHY-link interface

The evolutionary PHY-link interface is an 8 bit, parallel data interface that has been enhanced to allow data transfers at speeds of up to S800 and to support the additional request types required by IEEE Std 1394b-2002.

The evolutionary interface is specified as being more symmetrical, to avoid any deadlock or long latency situations where the link or PHY device is prevented from communicating with the other while a lengthy operation is taking place. This enhancement was made possible by inclusion of a unidirectional signaling line from the PHY to the link over which status information can be sent while the data bus is busy with packet data. Additionally, a clock is added at the link that is used when the data lines are carrying data from the link to the PHY. This additional clock allows source synchronous transfers in both directions and greatly simplifies the problems associated with moving data at high speeds.

In general, the PHY device is the owner of the PHY-link interface. All Serial Bus packets are transferred to the link device as they are received. The link requests the use of the Serial Bus via the PHY device. The PHY interprets and queues the link transmit requests as appropriate. When the PHY is granted access to the bus, the PHY passes that grant to the link, which then transmits data as required.

From a functional point of view, the PHY-link interface can be viewed as a master-slave interface. At any point in time, either the PHY or the link is the current owner of the interface, and a handoff mechanism to pass ownership from one to the other is defined.

3.10.6.2 Serial PHY-link interface

During the development of this standard, a great deal of effort was spent on trying to extend the capabilities of the Legacy PHY-link interface. It was evident that continuing to increase the transfer speeds without making the interface wider would be difficult. It would compromise the system design not only because of the increased pin counts on the PHY and the link devices, but also because of the implications to circuit board designs, especially designs that incorporated galvanic isolation between the PHY and the link. Additional problems were encountered in trying to accommodate future voltage scaling that might cause the PHY and the link to run on different supply voltages with different signal swings.

Because of these problems, a second PHY-link interface has been defined in this specification. This interface uses the standard Beta-mode signaling between the link and a fan-out PHY (FOP). The link, therefore, contains what is nearly the functional equivalent of a Beta-mode PHY. The PHY integrated into the link arbitrates directly for the Serial Bus by using standard BOSS arbitration, which eliminates the need for any additional signals (e.g., LREQ) between the PHY and the link.

The PHY that is integrated into the link (PIL) can be a simplified version of a standard Beta-only PHY because this PHY does not participate in the tree-ID or self-ID process. Additionally, because this PHY is always a leaf node, this PHY does not need to participate in the loop-free build process.

To preserve software compatibility with the Legacy model of the PHY-link interface, the PIL shares the node ID of the FOP. Additionally, a special packet format allows the link to transmit and receive PHY register data to and from the FOP over the two signaling pairs. This special format allows software to access the FOP as if it were attached through the Legacy PHY-link interface.

The new interface continues to have a LinkOn signal.

3.10.7 Miscellaneous features

3.10.7.1 Autonegotiation

IEEE Std 1394b-2002 has greatly enhanced the PHY state machines to use autonegotiation to determine the speed and mode of signaling to be used between nodes. Because nodes may be connected through optical media, there can be no assurance of seeing dc bias changes to indicate connection changes. IEEE Std 1394b-2002 solves the connection change problem by the use of tones, which are detected and acted upon in an handshake manner. Autonegotiation supports dc copper connections, ac-coupled copper, and purely optical connections. Beta mode is used if at all possible. DS mode is used for bilingual ports when a DS-only node is detected on the other side of the connection. Autonegotiation takes place before bus reset, tree identification, and self-identification occur.

3.10.7.2 Loop breaking

A new feature is loop breaking. Loops are automatically removed during initialization.

3.10.7.3 Power conservation improvements

IEEE Std 1394b-2002 adds standby and restore features for power conservation beyond the suspend and resume features of IEEE Std 1394a-2000. Standby state is restricted to leaf nodes, and bus resets do not occur when a port enters and comes out of the Standby state. When the PHY's port is in the Standby state, it may enter a low-power mode with power consumption similar to the consumption attained when in the Suspended state.

3.10.7.3.1 Standby state

Standby state is a feature only of Beta-mode operation. The term *nephew* is used to label a leaf node that has a port in the Standby state. The active node connected to a nephew node is an *uncle*.

Although a nephew does not participate in normal bus activity, the active bus, of which the nephew is a member, is not aware of a change in status of a node as it becomes a nephew, i.e., a bus reset does not occur. If a bus reset occurs while a nephew has a port in Standby state, then the uncle proxies self-ID packet(s) on behalf of a nephew.

A node becomes nephew when it detects a standby command packet containing its node ID and port address. A root node, a node with more than one active port, or a node that has another port already in Standby state cannot become a nephew.

3.10.7.3.2 Restore

The port in Standby state on the nephew restores from Standby state upon detecting a restore tone from its uncle. A nephew restores its port from Standby state when it receives any arbitration request from its link, i.e., the nephew's port asserts a restore tone to its uncle. Upon detecting or generating a restore tone, a nephew begins monitoring for a PHY packet. The PHY packet contains information used by the nephew to restore context, i.e., bus phase, node ID, gap count, and gap count sticky bit status.

A multiport node with one connected port in Standby state restores that port followed by a bus reset if, on any of its other ports, it detects a resume or a new connection.

4. Cable PHY specification

Insert the following after 4.2.1A, which was inserted into IEEE Std 1394-1995 by IEEE Std 1394a-2000:

4.2.1B Copper PMD cable media attachment

This specification defines a new media and connector to support the higher data rates defined in IEEE Std 1394b-2002. This media and connector are known as the Beta interface. This interface is usable at speeds up to the maximum defined by IEEE Std 1394b-2002, S1600. To enable interoperability with Legacy devices, a keyed version of the Beta connector is defined. This connector allows bilingual ports to be attached to either DS ports or to Beta-only ports. The Beta connector is used to prevent Legacy ports from being attached to Beta-only ports. An interconnect matrix is provided in Table 4-11J and Table 4-11K of this subclause.

The Beta interface has been designed to meet the small size requirements of consumer electronics devices while meeting the higher speed and power distribution needs of computer and peripheral devices.

The 4- and 6-circuit receptacles defined in IEEE Std 1394-1995 and IEEE Std 1394a-2000 shall not be used on a port that is capable of operating in Beta mode. Only the receptacles defined in this subclause are allowed for use for short-haul (< 5 m) copper connections on a Beta-capable port.

4.2.1B.1 Connectors

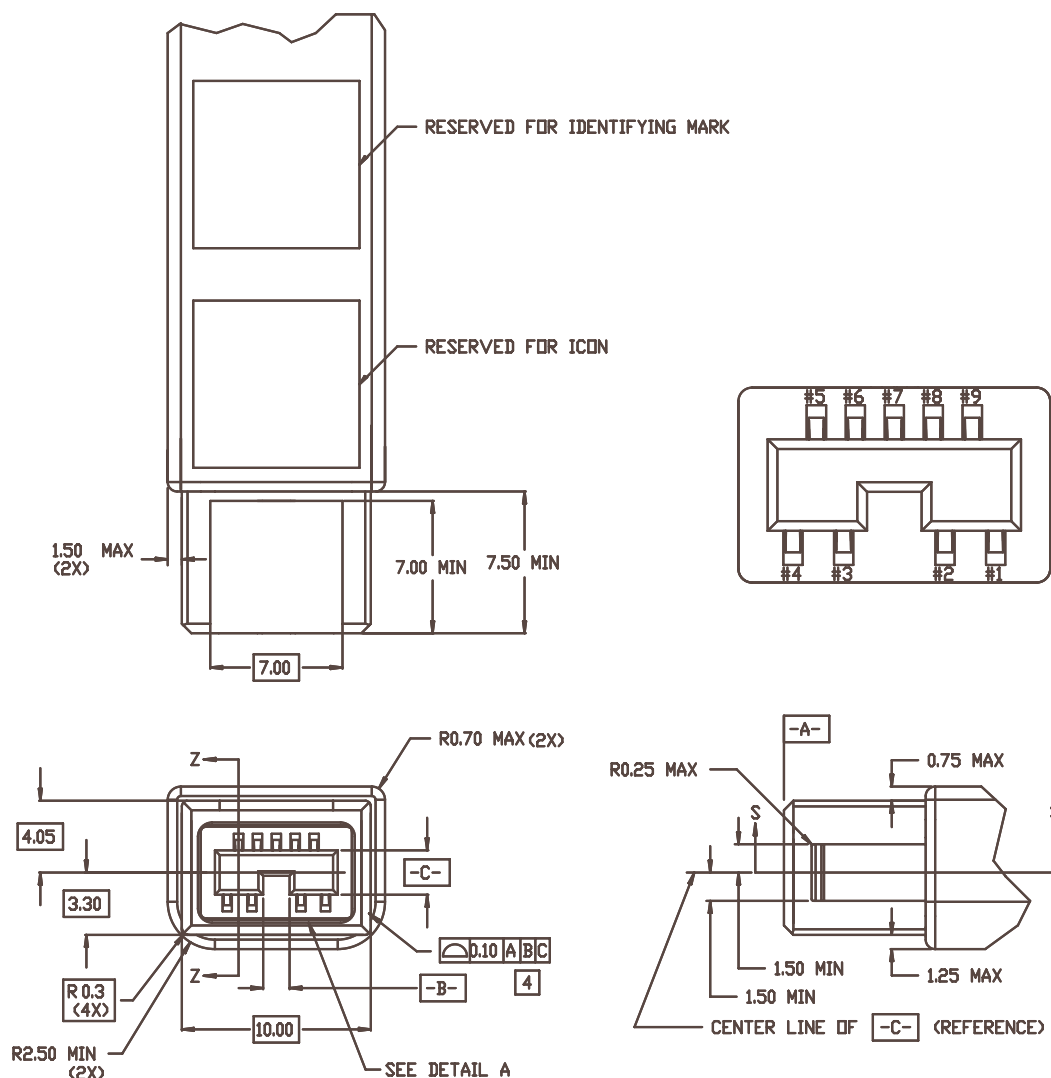
The connectors and cables defined in this subclause are used to interconnect B nodes at distances of up to 4.5 m. This subclause specifies the electrical and physical properties of the Beta and bilingual cables and connectors. Some of the connector attributes that are not directly specified in this subclause are implied by the performance requirements in 4.2.1B.3.

Information about the recommended planar mounting footprints of the sockets is found in Figure 4-11AE and Figure 4-11AF. Compliance to these footprints is optional, but performance of alternative footprints should be comparable to the footprints shown in this specification.

In typical computer applications, consumer electronic or peripheral equipment boxes present one or more sockets for attachment to other boxes via cables. The detachable ends of the cable shall be terminated with plugs. IEEE Std 1394b-2002 defines the copper Beta and bilingual cables and their mating printed circuit board (PCB) socket interfaces.

All dimensions, tolerances, and descriptions of features that affect the intermateability of the Beta and bilingual plugs and sockets are specified within this subclause. Connector features that are not directly controlled within this subclause are indirectly controlled by the performance requirements in 4.2.1B.3.

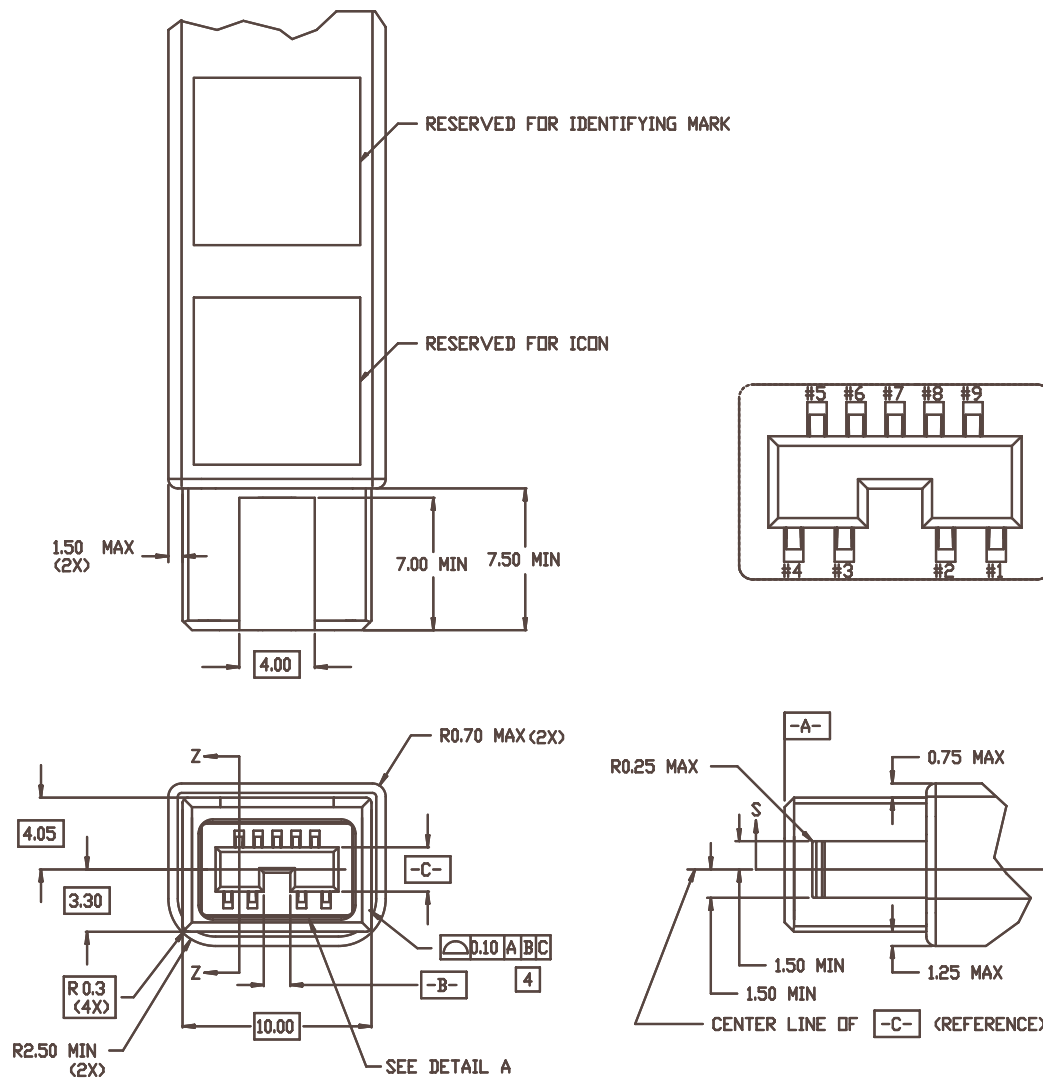
The mating features of the Beta plug and overmold features are specified in Figure 4-11N. The bilingual plug and overmold features are shown in Figure 4-11O. The interface, unmated plug cross-section, and detent common to both the Beta and bilingual plugs are found in Figure 4-11P, Figure 4-11Q, and Figure 4-11R. Adherence to these details helps ensure mating integrity by multiple manufacturers of these cable plugs.



NOTES:

- [1] All linear dimensions are in millimeters.
- [2] Unless otherwise specified, tolerances are linear ± 0.15 and angular $\pm 5^\circ$.
- [3] Interpret dimensions and tolerances per ANSI Y14.5M-1994.
- [4] Theoretical sharp corner, chamfer profile tolerance excludes radii.

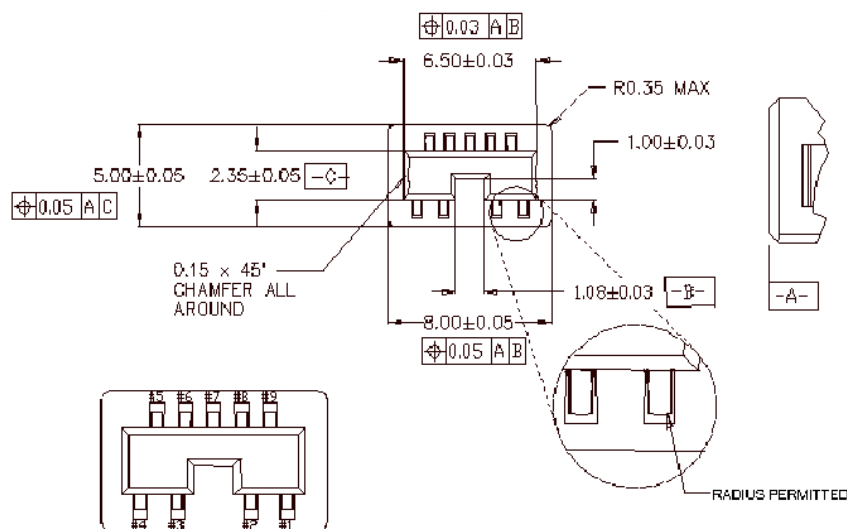
Figure 4-11N—Beta plug body with overmold



NOTES:

- [1] All linear dimensions are in millimeters.
- [2] Unless otherwise specified, tolerances are linear ± 0.15 and angular $\pm 5^\circ$.
- [3] Interpret dimensions and tolerances per ANSI Y14.5M-1994.
- [4] Theoretical sharp corner, chamfer profile tolerance excludes radii.

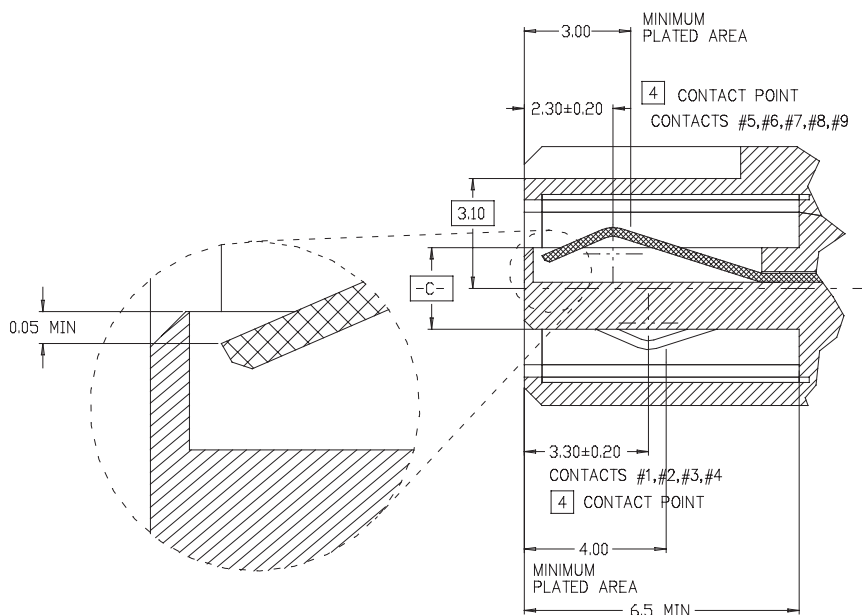
Figure 4-110—Bilingual plug body with overmold



NOTES:

- [1] All linear dimensions are in millimeters.
- [2] Unless otherwise specified, tolerances are linear ± 0.15 and angular $\pm 5^\circ$.
- [3] Interpret dimensions and tolerances per ANSI Y14.5M-1994.

Figure 4-11P—Beta and bilingual plug interface: Detail A



NOTES:

- [1] All linear dimensions are in millimeters.
- [2] Unless otherwise specified, tolerances are linear ± 0.15 and angular $\pm 5^\circ$.
- [3] Interpret dimensions and tolerances per ANSI Y14.5M-1994.
- [4] Contact shape is manufacturer's option.

Figure 4-11Q—Beta and bilingual plug section Z-Z (unmated)