

INTERNATIONAL STANDARD

**IEC
60488-1**

First edition
2004-07

IEEE 488.1™

**Higher performance protocol for the standard
digital interface for programmable
instrumentation –**

**Part 1:
General**



IEEE

Reference number
IEC 60488-1(E):2004
IEEE Std. 488.1(E)-2003
Documents
Document Sharing Hub

Publication numbering

As from 1 January 1997 all IEC publications are issued with a designation in the 60000 series. For example, IEC 34-1 is now referred to as IEC 60034-1.

Consolidated editions

The IEC is now publishing consolidated versions of its publications. For example, edition numbers 1.0, 1.1 and 1.2 refer, respectively, to the base publication, the base publication incorporating amendment 1 and the base publication incorporating amendments 1 and 2.

Further information on IEC publications

The technical content of IEC publications is kept under constant review by the IEC, thus ensuring that the content reflects current technology. Information relating to this publication, including its validity, is available in the IEC Catalogue of publications (see below) in addition to new editions, amendments and corrigenda. Information on the subjects under consideration and work in progress undertaken by the technical committee which has prepared this publication, as well as the list of publications issued, is also available from the following:

- **IEC Web Site** (www.iec.ch)

- **Catalogue of IEC publications**

The on-line catalogue on the IEC web site (www.iec.ch/searchpub) enables you to search by a variety of criteria including text searches, technical committees and date of publication. On-line information is also available on recently issued publications, withdrawn and replaced publications, as well as corrigenda.

- **IEC Just Published**

This summary of recently issued publications (www.iec.ch/online_news/justpub) is also available by email. Please contact the Customer Service Centre (see below) for further information.

- **Customer Service Centre**

If you have any questions regarding this publication or need further assistance, please contact the Customer Service Centre:

Email: custserv@iec.ch
Tel: +41 22 919 02 11
Fax: +41 22 919 03 00

INTERNATIONAL STANDARD

IEC 60488-1

First edition
2004-07

IEEE 488.1™

Higher performance protocol for the standard digital interface for programmable instrumentation –

Part 1: General

Copyright © IEEE 2004 — All rights reserved

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

No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

International Electrotechnical Commission, 3, rue de Varembé, PO Box 131, CH-1211 Geneva 20, Switzerland
Telephone: +41 22 919 02 11 Telefax: +41 22 919 03 00 E-mail: inmail@iec.ch Web: www.iec.ch

The Institute of Electrical and Electronics Engineers, Inc, 3 Park Avenue, New York, NY 10016-5997, USA
Telephone: +1 732 562 3800 Telefax: +1 732 562 1571 E-mail: stds-info@ieee.org Web: www.standards.ieee.org



Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия



CONTENTS

Foreword.....	4
IEEE Introduction.....	8
1. Overview.....	10
1.1 Scope.....	10
1.2 Object.....	11
1.3 Interface system overview	11
2. References.....	15
3. Definitions	15
3.1 General system terms	15
3.2 Units connected via the interface system	16
3.3 Signals and paths	16
4. Functional specifications.....	17
4.1 Functional partition.....	17
4.2 Notation used to specify interface functions.....	21
4.3 Source handshake (SH) interface function	24
4.4 Acceptor handshake (AH) and extended acceptor handshake (AHE) interface functions	32
4.5 Talker (T) interface function (Includes serial poll capabilities)	41
4.6 Listener (L) interface function	49
4.7 Service request (SR) interface function	55
4.8 Remote local (RL) interface function	57
4.9 Parallel poll (PP) interface function.....	60
4.10 Device clear (DC) interface function.....	64
4.11 Device trigger (DT) interface function	66
4.12 Controller (C) interface function	68
4.13 Remote message coding and transfer.....	80
4.14 Configuration (CF) interface function	86
5. Electrical specifications.....	91
5.1 Application.....	91
5.2 Logical and electrical state relationships	91
5.3 Driver requirements	91
5.4 Receiver requirements.....	92
5.5 Composite device load requirements	92
5.6 Ground requirements.....	94
5.7 Cable characteristics.....	95
5.8 State transition timing values	95
6. Mechanical specifications.....	98
6.1 Application.....	98
6.2 Connector type	98
6.3 Connector contact assignments.....	99
6.4 Device connector mounting	100
6.5 Cable assembly	101

7.	System applications and guidelines for the designer	103
7.1	System compatibility.....	103
7.2	Data rate consideration.....	103
7.3	Device capabilities	104
7.4	AND and OR logic operations	104
7.5	Address assignment.....	104
7.6	Typical combinations of interface functions.....	107
7.7	Unimplemented interface message handling	109
8.	System requirements and guidelines for the user	110
8.1	System compatibility	110
8.2	System installation requirements	110
8.3	Address assignment	111
8.4	Cabling restrictions	112
8.5	Operational sequence guidelines	112
Annex A (informative)	Typical instrument system	116
Annex B (informative)	Handshake process timing sequence.....	118
Annex C (informative)	Interface function-allowable subsets.....	124
Annex D (informative)	Interface message reference list	134
Annex E (informative)	Multiline interface messages: ISO code representation.....	138
Annex F (informative)	Logic circuit implementation.....	139
Annex G (informative)	Parallel polling sequence	141
Annex H (informative)	Description of interface parameters on data sheets.....	142
Annex I (informative)	Holdoff considerations for noninterlocked transfers	146
Annex J (informative)	Address switch labeling and interface status indicators	147
Annex K (informative)	Recommended methods for reducing the effects of radiated and conducted interference for devices specified in this standard.....	150
Annex L (informative)	List of Participants	153

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**HIGHER PERFORMANCE PROTOCOL FOR THE STANDARD DIGITAL
INTERFACE FOR PROGRAMMABLE INSTRUMENTATION –****Part 1: General**

FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.
- 6) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC/IEEE 60488-1 has been processed through subcommittee 65C: Digital communications, of IEC technical committee 65: Industrial-process measurement and control.

This standard cancels and replaces the second edition of IEC 60625-1 (1993).

At times in this standard, specific reference is made to IEEE Std 488.1-1987, which constituted an earlier version of IEEE Std 488.1-2003, the IEEE edition upon which this standard is based. Where specific dated references were made to the 1987 edition, these references have been maintained.

Furthermore, it is to be noted that full compatibility of this standard with IEC/IEEE 60488-2:2004 requires implementation of all revisions indicated previously in the IEEE Introduction. Therefore, readers of this standard are encouraged to read also the companion standard IEEE Std 488.2-1987, which constitutes an earlier edition of IEC 60488-2:2004.

The text of this standard is based on the following documents:

IEEE Std	FDIS	Report on voting
488.1 (2003)	65C/319A/FDIS	65C/343/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives.

The committee has decided that the contents of this publication will remain unchanged until 2009.

IEC/IEEE 60488 consists of the following publications:

- Higher performance protocol for the standard digital interface for programmable instrumentation – Part 1: General (60488-1).
- Standard digital interface for programmable instrumentation – Part 2: Codes, formats, protocols and common commands (60488-2).

IEC/IEEE Dual Logo International Standards

This Dual Logo International Standard is the result of an agreement between the IEC and the Institute of Electrical and Electronics Engineers, Inc. (IEEE). The original IEEE Standard was submitted to the IEC for consideration under the agreement, and the resulting IEC/IEEE Dual Logo International Standard has been published in accordance with the ISO/IEC Directives.

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 IEC/IEEE Dual Logo International Standard is wholly voluntary. The IEC and IEEE disclaim 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 IEC or IEEE Standard document.

The IEC and IEEE do not warrant or represent the accuracy or content of the material contained herein, and expressly disclaim 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. IEC/IEEE Dual Logo International Standards documents are supplied "AS IS".

The existence of an IEC/IEEE Dual Logo International 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 IEC/IEEE Dual Logo International 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 IEC and IEEE are not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Neither the IEC nor IEEE is undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEC/IEEE Dual Logo International Standards or 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 IEC/IEEE Dual Logo International Standards are welcome from any interested party, regardless of membership affiliation with the IEC or 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 and/or General Secretary, IEC, 3, rue de Varembe, PO Box 131, 1211 Geneva 20, Switzerland.

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.

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.

IEEE Standard for Higher Performance Protocol for the Standard Digital Interface for Programmable Instrumentation

Sponsor

Technical Committee on Automated Test Systems and Instrumentation (TC-8)
of the
IEEE Instrumentation and Measurement Society

Approved 10 October 2003
American National Standard Institute

Approved 12 June 2003
IEEE-SA Standards Board

Abstract: This standard applies to interface systems used to interconnect both programmable and nonprogrammable electronic measuring apparatus with other apparatus and accessories necessary to assemble instrumentation systems. It applies to the interface of instrumentation systems, or portions of them, in which the

- a) Data exchanged among the interconnected apparatus is digital (as distinct from analog)
- b) Number of devices that may be interconnected by one contiguous bus does not exceed 15
- c) Total transmission path lengths over the interconnecting cables does not exceed 20 m
- d) Data rate among devices does not exceed 8 000 000 B/s

The basic functional specifications of this standard may be used in digital interface applications that require longer distances, more devices, increased noise immunity, or combinations of these. Different electrical and mechanical specifications may be required (for example, symmetrical circuit configurations, high threshold logic, special connectors, or cable configurations) for these extended applications.

Keywords: GPIB, HPIB, HS488, non-interlocked handshake, three-wire handshake

IEEE Introduction

IEEE Std 488™ has enjoyed continuous and widespread use since its initial publication in 1975. The first revision occurred in 1978 as a result of practical experience and recognition that certain clauses needed clarification to improve compatibility among independently designed products. No major changes were made in 1978; many changes were pure editorial; however, 20 clauses had textual changes with technical implications, although none contradicted the concepts as defined in the original publication. Supplement A was introduced in 1980 to correct one minor deficiency in the controller function related to “take control synchronously.”

In 1987, a systematic review was undertaken as a result of both the normal 5-year review cycle and related work on IEEE 488 device-dependent message syntax structures. In addition, there was a strong desire on the part of both IEEE participants and our IEC colleagues to bring equivalent standards (IEC 625-1) into closer alignment. IEEE Std 488.1-1987 represents the culmination of this review cycle. Again, no major technical changes were made, and care was exercised to preserve compatibility with earlier versions of IEEE Std 488.

The IEEE Std 488.1-2003 specification adds new interface functions to allow designers the option of implementing noninterlocked handshake transfers.

In preparing this specification, several stylistic changes to IEEE Std 488.1-1987 were necessary to bring the document up to IEEE standards. The following changes were made:

- Clause 1, which previously consisted of an overview, definitions, and references, was divided into three separate sections. Clause 1 now contains the overview, Clause 2 contains references, and Clause 3 contains definitions.
- The first level of ordered lists was previously numbered (1, 2, 3...). To comply with the IEEE standards, the first level of ordered lists is now lettered (a, b, c...), and the second level of ordered lists is numbered (1, 2, 3...).
- In the earlier version of this specification, tables had no grids. Grids have been added to the tables in this version.

The IEEE 488.1-2003, IEEE Standard for Digital Interface for Programmable Instrumentation, deals with systems that use a byte-serial, bit-parallel means to transfer digital data among a group of instruments and system components. The interface system described herein is optimized as an interdevice interface for system components in relatively close proximity able to communicate over a contiguous party-line bus system.

This document contains seven sections as follows:

- Clause 1 contains the scope, the object, and summary description of the interface.
- Clause 2 contains helpful references.
- Clause 3 contains basic definitions.
- Clause 4 deals with functional concepts and specifications of the interface system described in this standard. One or more interface functions contained within a device are each able to process messages and change states to maintain an orderly flow of information among a set of interconnected devices.
- Clause 5 deals with the electrical realization of the interface in order to transfer messages among a set of interconnected devices.
- Clause 6 deals with the mechanical realization of the interface in order to implement the electrical aspects of the interface system.
- Clause 7 deals with system considerations that must be given to the design of an individual device in order to make it compatible with other devices of a measurement system.

- Clause 8 deals with system considerations that must be recognized by the user of devices designed in accordance with this standard.
- Annexes deal with explanatory matter and examples.

In order to interconnect and program equipment designed in accordance with this standard, the user should have knowledge of Clause 1 and Clause 8. If the coding and transfer of messages is not done automatically by the apparatus to be programmed, it will be necessary that the user have knowledge of Clause 4. General familiarity with the other sections is recommended. The user must also be familiar with device-dependent characteristics of apparatus that may be used in a system, but that are beyond the scope of this standard.

This standard defines an interface with the objective to assure that messages may be accurately communicated between two or more devices in a system, but it does not guarantee that each device will interpret properly all possible messages sent to it or will properly generate all necessary messages. A wide latitude of interface capability is permitted within the scope of this standard, which may permit operational incompatibility among interconnected devices.

Device designers must have sufficient awareness of the characteristics of systems, which might include their devices, in order to select correctly among the options provided in this standard. Likewise, system configurators must have sufficient awareness of the options included in each of the devices in their systems in order to ensure that the correct communication techniques are used.

This standard does not specify the device-dependent or operational characteristics required for complete system compatibility. Therefore, following the rules and procedures of this standard alone will not guarantee unconditional compatibility.

The 1987 version of this standard was based on work initiated by the International Electrotechnical Commission (IEC) within Technical Committee 65, Subcommittee 65C, Working Group 3 (formerly TC66/WG3), and it follows the general concepts of a standard prepared by the IEC. In 1992, the IEEE technical working group enhanced the 1987 standard to improve performance over IEEE Std 488.1-1987.

The “helpful note” on metric threads found in previous editions has been deleted because metric thread use is common IEEE Std 488 practice. Consequently, the recommendation to coat such parts in black material to call attention to metric threads is also considered unnecessary. Electrical conductivity on the surface of these parts is, however, still considered essential.

Patents

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents 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. A patent holder has filed a statement of assurance that it will grant licenses under these rights without compensation or under reasonable rates and nondiscriminatory, reasonable terms and conditions to all applicants desiring to obtain such licenses. The IEEE makes no representation as to the reasonableness of rates and/or terms and conditions of the license agreements offered by patent holders. Further information may be obtained from the IEEE Standards Department.

Interpretations and errata

Interpretations and errata associated with this standard may be found at one of the following Internet locations:

- <http://standards.ieee.org/reading/ieee/interp/>
- <http://standards.ieee.org/reading/ieee/updates/errata/>

HIGHER PERFORMANCE PROTOCOL FOR THE STANDARD DIGITAL INTERFACE FOR PROGRAMMABLE INSTRUMENTATION –

Part 1: General

1. Overview

1.1 Scope

This standard applies to interface systems used to interconnect both programmable and nonprogrammable electronic measuring apparatus with other apparatus and accessories necessary to assemble instrumentation systems. It applies to the interface of instrumentation systems, or portions of them, in which the

- a) Data exchanged among the interconnected apparatus is digital (as distinct from analog)
- b) Number of devices that may be interconnected by one contiguous bus does not exceed 15
- c) Total transmission path lengths over the interconnecting cables does not exceed 20 m
- d) Data rate among devices does not exceed 8 000 000 B/s

The basic functional specifications of this standard may be used in digital interface applications that require longer distances, more devices, increased noise immunity, or combinations of these. Different electrical and mechanical specifications may be required (for example, symmetrical circuit configurations, high threshold logic, special connectors, or cable configurations) for these extended applications.

This standard may also be applicable to other instrumentation system elements, such as processors, stimulus, display, or storage devices, and terminal units found useful in instrumentation systems. It applies generally to laboratory and production test environments that are both electrically quiet and restricted as to physical dimensions (distances between the system components).

This standard deals only with the interface characteristics of instrumentation systems to the exclusion of design specifications' consideration of radio-interface regulations, performance requirements, and safety requirements of apparatus.

NOTE—For the latter two items, reference is made to IEC 61010-1: 2001, and IEC 60359:2001.¹

¹For information on references, see Clause 2.

Throughout this standard, and insofar as further distinction is not necessary, the term “system” denotes the bit-parallel byte-serial interface system that, in general, includes all circuits, cables, connections, message repertoire, and control protocol to effect unambiguous data transfer between devices; and the term “device” or “apparatus” denotes any programmable measurement device or other product connected to the interface system that communicates information via, and conforms to, the interface system definition.

A primary focus of this standard is to set forth an interface system to interconnect self-contained apparatus to other apparatus by external means. This same standard may be applied to interconnecting the internal sub-sections within a self-contained equipment.

1.2 Object

This standard is intended

- a) To define a general-purpose system for use in limited-distance applications
- b) To specify the device-independent mechanical, electrical, and functional interface requirements that the apparatus shall meet in order to be interconnected and to communicate unambiguously via the system
- c) To specify the terminology and definitions related to the system
- d) To enable the interconnection of independently manufactured apparatus into a single functional system
- e) To permit devices with a wide range of capability—from the simple to the complex—to be interconnected to the system simultaneously
- f) To permit direct communication among the devices without requiring all messages to be routed to a control or intermediate unit
- g) To define a system with a minimum of restrictions on the performance characteristics of the devices connected to the system
- h) To define a system that permits asynchronous communication over a wide range of data rates
- i) To define a system that, of itself, may be relatively low cost and permits the interconnection of low-cost devices
- j) To define a system that is easy to use

1.3 Interface system overview

1.3.1 Interface system objective

The overall purpose of an interface system is to provide an effective communication link over which messages are carried in an unambiguous way among a group of interconnected devices.

Messages (quantities of information) carried by an interface system belong to either of two broad categories:

- a) Messages used to manage the interface system itself, hereinafter called interface messages
- b) Messages used by the devices interconnected via the interface system that are carried by, but not used or processed by, the interface system directly, hereinafter called device-dependent messages

NOTE—The detailed specification of device-dependent messages is beyond the scope of this standard.

1.3.2 Fundamental communication capabilities

An effective communication link requires three basic functional elements to organize and manage the flow of information to be exchanged among devices:

- a) A device acting as a listener
- b) A device acting as a talker
- c) A device acting as a controller

In the context of the interface system described by this standard

- a) A device with the capability to listen can be addressed by an interface message to receive device-dependent messages from another device connected to the interface system.
- b) A device with the capability to talk can be addressed by an interface message to send device-dependent messages to another device connected to the interface system.
- c) A device with the capability to control can address other devices to listen or to talk. In addition, this device can send interface messages to command specified actions within other devices. A device with only this capability neither sends nor receives device-dependent messages.

NOTE—The use of the word *controller* throughout this standard applies strictly to the management (control) of the interface system and does not imply the broad capabilities typically associated with the word in the data processing context. Further classification of the controller will be made in Clause 4 to distinguish between different types of controller capabilities related to the interface system.

Listener, talker, and controller capabilities occur individually or in any combination in devices interconnected via the interface system, as shown in Figure 1.

In addition to the basic listener, talker, and controller functions, the system provides interface messages to accomplish the following operations:

- a) A serial poll sequence may be initiated when a device (with talker function) requires some action by the controller, by transmitting the service request message. The controller will then obtain the status byte of all possible devices in sequence to ascertain which required service.
- b) The Parallel Poll function provides a device with the ability to transmit on the controller's demand one bit of status information (request service) simultaneously with several other devices. The assignment of a data line to a particular device for the response to a parallel poll may be accomplished through interface messages.
- c) The Device Clear and Device Trigger functions provide a device with the ability to be initialized or triggered, respectively, on command from the controller. This may occur simultaneously with other selected or all devices in a system.
- d) The remote/local function provides a device with the ability to accept program data from the bus, local data (for example, front panel controls), or both.

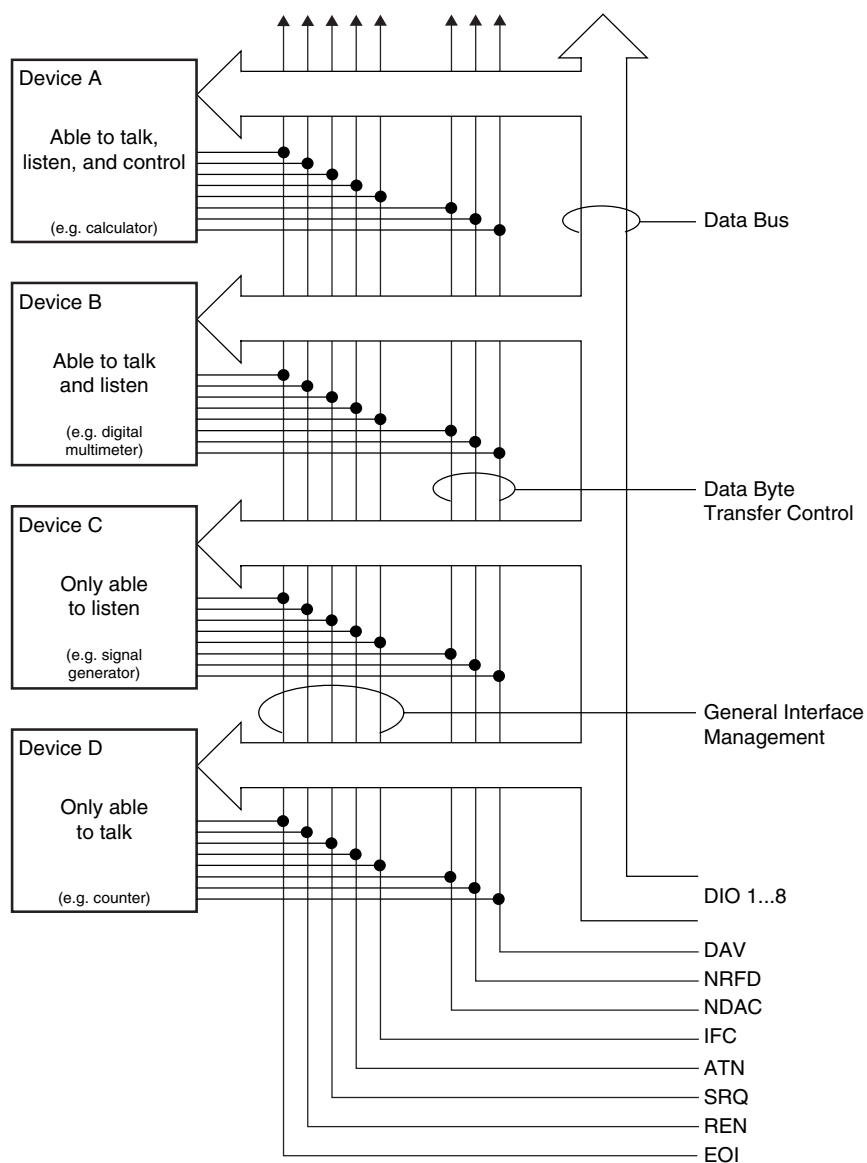


Figure 1—Interface capabilities and bus structure

1.3.3 Message paths and bus structure

The interface system contains a set of 16 signal lines used to carry all information, interface messages, and device-dependent messages among interconnected devices.

Messages may be coded on one or a set of signal lines as determined by the particular message content and its relationship to the interface system.

The bus structure is organized into three sets of signal lines:

- a) Data bus, eight signal lines
- b) Data byte transfer control bus, three signal lines
- c) General interface management bus, five paths

Figure 1 illustrates the basic communication paths.

A set of eight interface signal lines carries all 7 bit interface messages and the device-dependent messages

DIO1 (data input output 1)

·
·
·
·

DIO8 (data input output 8)

Message bytes are carried on the DIO signal lines in a bit-parallel byte-serial form, asynchronously, and generally in a bidirectional manner.

NOTE—A message may be carried on an individual DIO signal line when required.

A set of three interface signal lines is used to effect the transfer of each byte of data on the DIO signal lines from a talker or controller to one or more listeners:

- a) Data Valid (DAV) is used to indicate the condition (availability and validity) of information on the DIO signal lines.
- b) Not Ready For Data (NRFD) is used to indicate the condition of readiness of device(s) to accept data or (by a source) to indicate to all acceptors that it is capable of supporting noninterlocked handshake cycles.
- c) Not Data Accepted (NDAC) is used to indicate the condition of acceptance of data by device(s).

The DAV, NRFD, and NDAC signal lines operate in what is called a three-wire (interlocked) handshake or a noninterlocked handshake process to transfer each data byte across the interface.

Five interface signal lines are used to manage an orderly flow of information across the interface:

- a) Attention (ATN) is used (by a controller) to specify how data on the DIO signal lines are to be interpreted and which devices must respond to the data.
- b) Interface Clear (IFC) is used (by a controller) to place the interface system, portions of which are contained in all interconnected devices, in a known quiescent state.
- c) Service Request (SRQ) is used by a device to indicate the need for attention and to request an interruption of the current sequence of events.
- d) Remote Enable (REN) is used (by a controller), in conjunction with other messages, to enable or disable one or more local controls that have corresponding remote controls.
- e) End or Identify (EOI) is used (by a talker) to indicate the end of a multiple byte transfer sequence or, in conjunction with ATN (by a controller), to execute a polling sequence.

1.3.4 Interface system elements

The primary elements of this interface system are as follows:

- a) Functional elements
- b) Electrical elements
- c) Mechanical elements

Each is described in a following clause.

2. References

This standard shall be used in conjunction with the following standards. For this standard, all references have been updated to reflect the most recent editions. When these references have been superseded by an approved revision, the revision shall apply. For undated references, the latest edition of the referenced document (including any amendments) applies.

ANSI X3.4-1986, American National Standard Code for Information Interchange Coded Character Set—7-Bit.²

IEC 60068-2 (all parts) Environmental Testing – Part 2: Tests.³

IEC 61010-1:2001, Safety Requirements for Electrical Equipment for Measurement, Control, and Laboratory Use – Part 1: General requirements.

IEC 60359:2001: Electrical and Electronic Measuring Equipment — Expression of Performance.

MIL STD 202F (1986), Test Method for Electronic and Electrical Component Parts.⁴

3. Definitions

For the purposes of this recommended practice, the following terms and definitions apply. *The Authoritative Dictionary of IEEE Standards Terms*, Seventh Edition, should be referenced for terms not defined in this clause.

3.1 General system terms

3.1.1 compatibility: The degree to which devices may be interconnected and used, without modification, when designed as defined throughout this standard (for example, mechanical, electrical, or functional).

3.1.2 handshake cycle: The process whereby digital signals effect the transfer of each data byte across the interface by means of a sequence of status and control signals. It may be interlocked or noninterlocked. Interlocked denotes a fixed sequence of events in which one event in the sequence must occur before the next event may occur.

3.1.3 interface: A common boundary between a considered system and another system, or between parts of a system, through which information is conveyed.

3.1.4 interface system: The device-independent mechanical, electrical, and functional elements of an interface necessary to effect communication among a set of devices. Cables, connector, driver and receiver circuits, signal line descriptions, timing and control conventions, and functional logic circuits are typical interface system elements.

3.1.5 local control: A method whereby a device is programmable by means of its local (front or rear panel) controls in order to enable the device to perform different tasks. (Also referred to as manual control.)

3.1.6 programmable: That characteristic of a device that makes it capable of accepting data to alter the state of its internal circuitry to perform a specific task(s).

3.1.7 remote control: A method whereby a device is programmable via its electrical interface connection, enabling it to perform different tasks.

²ANSI publications can be obtained from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA.

³IEC publications also are available in the US from the Sales Department, American National Standards Institute, 11 West 42nd Street, 113 Floor, New York, NY 10036, USA.

⁴MIL publications are available from Customer Service, Defense Printing Service, 700 Robbins Ave., Bldg. 4D, Philadelphia, PA 19111-5094.

3.1.8 system: A set of interconnected elements constituted to achieve a given objective by performing a specified function.

3.2 Units connected via the interface system

3.2.1 programmable measuring device: A measuring device that performs specified operations on command from the system and may transmit the results of the measurement(s) to the system.

3.2.2 terminal unit: A device that terminates the considered interface system and by means which a connection (and translation, if required) is made between the considered interface system and another external interface system.

3.3 Signals and paths

3.3.1 bidirectional bus: A bus used by any individual device for two-way transmission of messages, that is, both input and output.

3.3.2 bit-parallel: Refers to a set of concurrent data bits present on a like number of signal lines used to carry information. These data bits may be acted upon concurrently as a group (byte) or independently as individual data bits.

3.3.3 bus: A signal line or a set of signal lines used by an interface system to which a number of devices are connected and over which messages are carried.

3.3.4 byte: A group of adjacent binary digits operated on as a unit and usually shorter than a computer word (frequently connotes a group of eight bits).

3.3.5 byte-serial: A sequence of bit-parallel data bytes used to carry information over a common bus.

3.3.6 high state: The relatively more positive signal level used to assert a specific message content associated with one of two binary logic states.

3.3.7 low state: The relatively less positive signal level used to assert a specific message content associated with one of two binary logic states.

3.3.8 signal: The physical representation of information.

NOTE—For the purpose of this standard, this is a restricted definition of what is often called “signal” in more general terms, and it is hereinafter referred to digital electrical signals only.

3.3.9 signal level: The magnitude of signal compared to an arbitrary reference magnitude (voltage in the case of this standard).

3.3.10 signal line: One of a set of signal conductors in an interface system used to transfer messages among interconnected devices.

3.3.11 signal parameter: That parameter of an electrical quantity whose values or sequence of values convey information.

3.3.12 unidirectional bus: A bus used by any individual device for one-way transmission of messages only, that is, either input only or output only.

4. Functional specifications

4.1 Functional partition

A device is a physical entity designed for a particular application. It may be partitioned conceptually into three major functional areas each containing unique capabilities:

- a) Device functions (definition is application dependent)
- b) Interface functions (definition is application independent)
- c) Message coding logic

All communication to or from interface functions is defined in terms of messages and state linkages (see 4.1.4).

All messages carried on the signal lines are coded according to the coding logic defined in 4.13.

4.1.1 Device functions

The scope, purpose, size, content, and organization of the device function area (for example, analog signal measurement capability, range, modes of operation, etc.) are beyond the scope of this standard. Figure 2 illustrates the device function area B for which the designer has complete freedom to define device-related capability and the interface function area A for which the designer has no freedom to define new capability beyond that specified in this standard.

4.1.2 Interface function concepts

4.1.2.1 Interface functions

An interface function is the system element that provides the basic operational facility through which a device can receive, process, and send messages. A number of interface functions, each of which acts in accordance with specific protocol, are defined throughout this section of the standard. Each specific interface function may only send or receive a limited set of messages within particular classes of messages.

4.1.3 Interface function state

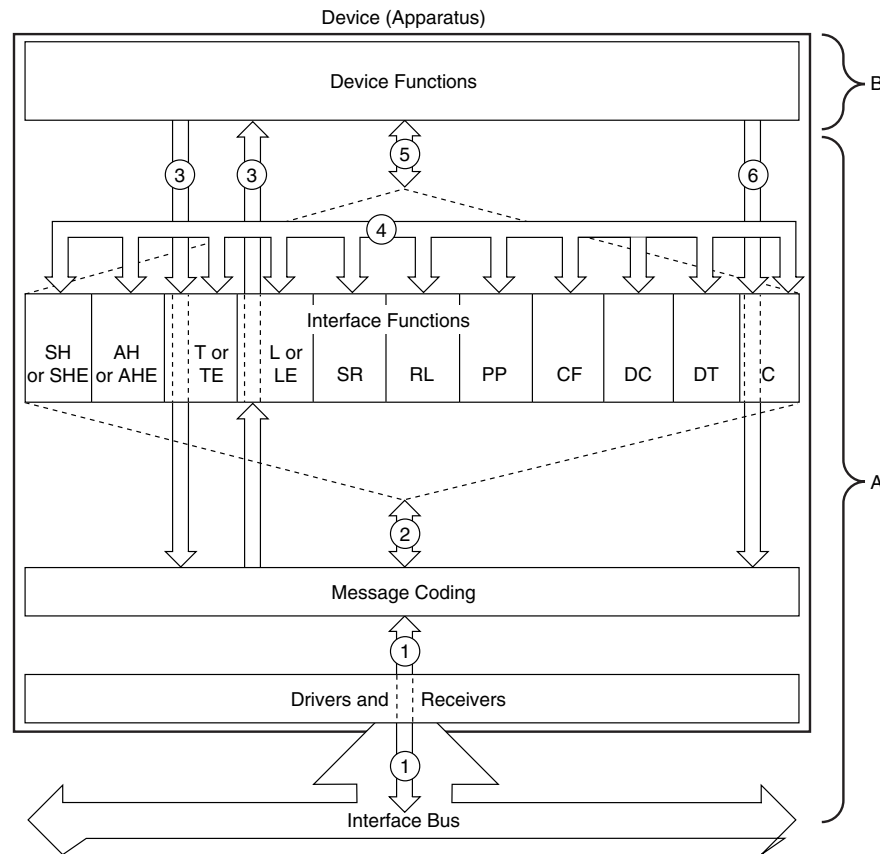
Each of the interface functions is defined in terms of one or more groups of interconnected, mutually exclusive states.

One and only one state shall be active at any one time within a single group of interconnected, mutually exclusive states.

For each state of an interface function, definitions are given for the following:

- a) Messages that may or must be sent over the interface while that state is active
- b) Conditions under which the function must leave that state and enter one of the other states in its group

These messages and conditions define the processing capability of the state.



- A = Capability defined by this standard
 B = Capability defined by the designer
 1 = Interface bus signal lines
 2 = Remote interface messages to and from interface functions
 3 = Device dependent messages to and from device functions
 4 = State linkages between interface functions
 5 = Local messages between device functions and interface functions
 (messages to interface functions are defined, messages from interface functions exist according to the designer's choice)
 6 = Remote interface messages sent by device functions within a controller

Figure 2—Functional partition within a device

4.1.3.1 Interface function repertoire

The designer is given the choice to select the particular set of interface functions necessary to fit the particular device application area. Figure 2 and Table 1 identify the available interface functions.

The total processing capability of a set of interface functions (designer selected set included in a specific device) at any moment is the logic conjunction of the processing capabilities of all those states (within each individual interface function) that are active at that moment.

4.1.3.2 Interface function assumptions and perspective

The state diagrams used to define the interface functions do not indicate, explicitly or implicitly, the intended existence of specific circuits elements to achieve the logical and physical implementation of a

function. For example, not all states necessarily imply the existence of a latched flip-flop or other memory elements.

The state diagrams used to define the interface functions are intended to permit the use of a wide variety of logic circuit implementations (for example, random logic, sequential logic, etc.).

The designer is free to combine and implement two or more interface functions with one logic design, provided all the conditions for each state of each interface function as defined in this section are met.

Throughout this section of the standard, the state diagrams, written descriptions, requirements, and guidelines are written for and should be interpreted from the device perspective. Clause 5 and Clause 6 will describe the interaction among devices from the system perspective.

An interface function must ignore (not respond to) any message coding not specifically defined.

A function may stay in any state for any amount of time (including zero) after exit conditions are met if this is not in conflict with specified constraints.

Table 1—Interface function repertoire

Interface Function	Symbol	Relevant Message Paths
Source handshake or extended source handshake	SH or SHE	1, 2, 4, 5
Acceptor handshake or extended acceptor handshake	AH or AHE	1, 2, 4, 5
Talker or extended talker	T or TE	1, 2, 3, 4, 5
Listener or extended listener	L or LE	1, 2, 3, 4, 5
Service Request	SR	1, 2, 4, 5
Remote local	RL	1, 2, 4, 5
Parallel Poll	PP	1, 2, 4, 5
Device Clear	DC	1, 2, 4, 5
Device Trigger	DT	1, 2, 4, 5
Controller	C	1, 2, 4, 5, 6
Configuration	CF	1, 2, 4, 5

4.1.4 Message concepts

4.1.4.1 Message

Each message represents a quantity of information and will be received either true or false at any specific time. All communications between an interface function and its environment are accomplished through messages sent or received.

4.1.4.2 Local message route and content

Messages sent between a device function and an interface function are called local messages.

Local messages flow between device functions and interface functions; see Figure 2, message route 5.

NOTE—Certain local messages are conveyed as remote messages and vice versa.

The designer is not allowed to introduce new local messages to interface functions.

The designer is allowed to introduce a local message derived from any state of any interface function to device function(s).

Local messages sent by device functions must exist for enough time to cause the required state transitions.

4.1.4.3 Remote message route and content

Messages sent via the interface between interface functions of different devices are called remote messages.

Each remote message is either an interface message or a device-dependent message.

Each interface message is sent to cause a state transition within another interface function. An interface message will not be passed along to the device when received by an interface function as shown in Figure 2, message route 2.

Device-dependent messages are passed between the device functions and the message coding logic via specified interface functions. These will cause no state transitions within the interface functions. Examples of device-dependent messages include device programming data, device measurement data, and device status data as shown in Figure 2, message route 3.

4.1.4.4 State linkage route and content

A state linkage is the logical interconnection of two interface functions where the transition to an active state of one interface function is dependent on the existence of a specified active state of another interface function, as indicated in Figure 2, message route 4.

4.1.4.5 Message coding

Message coding is the act of translating remote messages to or from interface signal line values. A message sent over a single line is called a uniline message. Two or more of these messages can be sent concurrently. A message that shares a group of signal lines with other messages, in some mutually exclusive set, is called a multiline message. Only one multiline message (message byte) can be sent at one time.

4.1.4.6 Classification of multiline messages

Multiline messages are interpreted as interface messages when the ATN message is true. Multiline messages are interpreted as device-dependent messages when the ATN message is false. The ATN message, when true, enables the accepting and processing of these specific classes of multiline messages:

- a) Universal commands (all devices)
- b) Addressed commands (all devices addressed to listen)
- c) Addresses (all devices)
- d) Secondary addresses or commands (all devices enabled by a primary address or command)

For a list of specific multiline commands, see Table 42.

4.1.4.7 Message transfer conventions

4.1.4.7.1 Remote message transfer conventions

- a) The value (true or false) of all remote false messages capable of being sent by a device shall at all times be as dictated by active states of its interface functions.
- b) The interface signal line(s) used to send a message value shall be set to the levels specified by Table 44.
- c) As normal interface operation allows two devices to simultaneously send opposite values of the remote messages, a technique shall be provided for resolving these conflicts. This is accomplished by implementing two types of message transfer over the interface, active transfer and passive transfer. The interface is structured so that in all conflicts between two message values, one of them will be active and the other will be passive. Messages shall be transferred in such a way that the active value overrides the passive value in every conflict that arises.
- d) A remote message can be transferred in one of four ways:
 - 1) An active true value being sent is guaranteed to be the value received, and the device need not allow it to be overridden.
 - 2) A passive true value being sent is not guaranteed to be the value received, and the device must allow it to be overridden.
 - 3) An active false value being sent is guaranteed to be the value received, and the device need not allow it to be overridden.
 - 4) A passive false value being sent is not guaranteed to be the value received, and the device must allow it to be overridden.
- e) Throughout the text, the terms true and false if not qualified are assumed to mean active true and active false during all discussions of remote message values sent by an interface function.
- f) For two specific remote messages, DAC and RFD, only false values are defined to be sent actively. Thus, an AND operation can be considered to be performed on the interface signal lines (see 7.4).
- g) For one remote message, SRQ, only true values are defined to be sent actively. Thus, an OR operation can be considered to be performed on the interface signal lines (see 7.4).
- h) Only the multiline message(s) to be sent true will be specified for an interface function state because multiline messages (sent via the DIO lines) are by their nature mutually exclusive. It should be understood that all unspecified multiline messages are sent passive false while the state is active.

4.1.4.7.2 Local message transfer conventions

- a) The coding of local messages is beyond the scope of this standard and is left to the discretion of the device designer.
- b) It is recommended that local messages qualifying transitions within any group of mutually exclusive states of an interface function be themselves mutually exclusive.

4.2 Notation used to specify interface functions

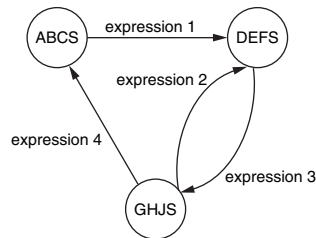
4.2.1 State diagram notation

Each state that an interface function can assume is represented graphically as a circle. A four-character upper-case alphanumeric mnemonic always ending in an S is used within the circle to identify the state:



All permissible transitions between states of an interface function are represented graphically by arrows between them.

Each transition is qualified by an expression whose value is either true or false. The interface function shall remain in its current state if all expressions that qualify transitions leading to other states are false. The interface function shall enter the state pointed to if, and only if, one of these expressions becomes true. The new state may be entered at any time after the expression(s) become(s) true, unless a time value is specified.



An expression consists of one or more local messages, remote messages, state linkages, or minimum time limits used in conjunction with the operators AND, OR, or NOT.

A local message to an interface function is represented by a three-letter mnemonic written in lower case, for example, rdy.

A remote message (received via the interface) is represented by a three-letter mnemonic written in upper case, for example, ATN. The representation may be appended by an integer; for example, PPR8.

A linkage from another state diagram is represented by a four-letter, bold, italicized mnemonic, for example, ***LACS***. A state linkage is true if the enclosed state is currently active; otherwise, it is false.

A minimum time limit is represented by the symbol T_n . This symbol achieves a true value only after the interface has been in the state originating the corresponding transition for the time value specified. It will remain true until the state is exited. The values for these time limits are contained in Table 48.

The AND operator is represented by the symbol $\wedge$.

The OR operator is represented by the symbol $\vee$.

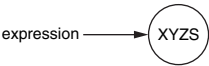
The AND operator takes precedence over the OR operator within an expression unless otherwise specified by parentheses.

The NOT operator is represented by a horizontal bar placed over the portion of the expression to be negated. The resulting negated expression has a true value if and only if the value of the expression under the bar is false.

If a transition is further qualified by a maximum time limit (within t_n), then the state pointed to shall be entered within the specified amount of time after the expression becomes true. The values for these time limits are contained in Table 48.

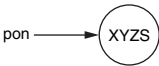
If a portion of an expression is optional in that its true value is not required for the complete expression to be true (at the designer's choice), then it is enclosed within square brackets [. . .].

If a specific expression causes a transition to a state from all other states of the diagram, shorthand notation is used instead of all the individual transitions being drawn. An arrow without a state at its origin is used to represent this condition, and it is assumed to originate in all states (for example, IFC or REN):

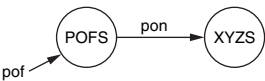


Although power-off (POFS) is a valid state of most interface functions and should normally be shown on all diagrams with a transition leading to the state to be entered at power-on time (pon), a shorthand form is used showing the pon pseudomessage originating a transition to the first state to be entered when power is turned on:

- a) Abbreviated notation used on state diagram:



- b) Complete representation implied by preceding symbol:



4.2.2 Message output notation

The message output table included with each interface function state diagram summarizes only the remote messages allowed to be sent during each of the states of the function.

Rows of the table are used to indicate states of the interface function.

Columns of the table are used to indicate remote messages allowed to be sent during at least one state of the interface function.

Each table entry indicates the value of a message that shall be sent while a specified state is active:

- a) T indicates active true
- b) F indicates active false
- c) (T) indicates passive true
- d) (F) indicates passive false

One column in each table is allocated, if required, to the group of multiline remote messages allowed to be sent. The multiline message to be sent true during each state is placed in its corresponding table entry. False values are not shown because multiline messages are mutually exclusive. Parentheses around a multiline message name specify that it shall be sent passive rather than active true.

A separate column for device function interaction summarizes the corresponding types of messages (or resultant action) that device function(s) are allowed to send or receive. Local message(s), beyond the scope of this standard, from the interface function to the device functions may be used to coordinate the appropriate action at the choice of the designer.

4.3 Source handshake (SH) interface function

4.3.1 General description

The SH interface function provides a device with the capability to guarantee the proper transmission of multiline messages. Two alternative versions of the function exist: the SH function and the extended source handshake (SHE) function. The SHE interface function is a superset of the SH interface function. Only one of the two alternative functions shall be implemented in a specific device.

The SH function, or SHE function, controls the initiation of, and termination of, the transmission of a multiline message byte. This function utilizes the data available (DAV), ready for data (RFD), and data accepted (DAC) messages to effect message byte transfers.

Transfers between an SHE function and an extended acceptor handshake (AHE) function may use noninterlocked handshake cycles. Other transfers use interlocked handshake cycles.

NOTE—Both the SH function and the SHE function are described concurrently throughout 4.3 due to the extensive similarity between these two functions.

4.3.2 SH function state diagram

The SH function shall be implemented to perform according to the state diagram given in Figure 3 and the state descriptions given throughout 4.3. Table 2 specifies the set of messages and states required to effect transition from one active state to another. Table 4 specifies the messages that shall be sent and the device function interaction required while each state is active.

The SHE function shall be implemented to perform according to the state diagram given in Figure 4 and the state descriptions given throughout 4.3. Table 3 specifies the set of messages and states required to effect transition from one active state to another. Table 5 specifies the messages that shall be sent and the device function interaction required while each state is active.

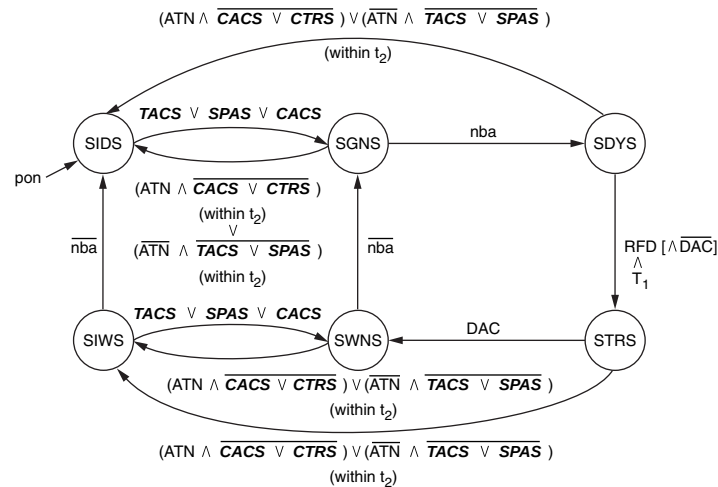


Figure 3—SH state diagram

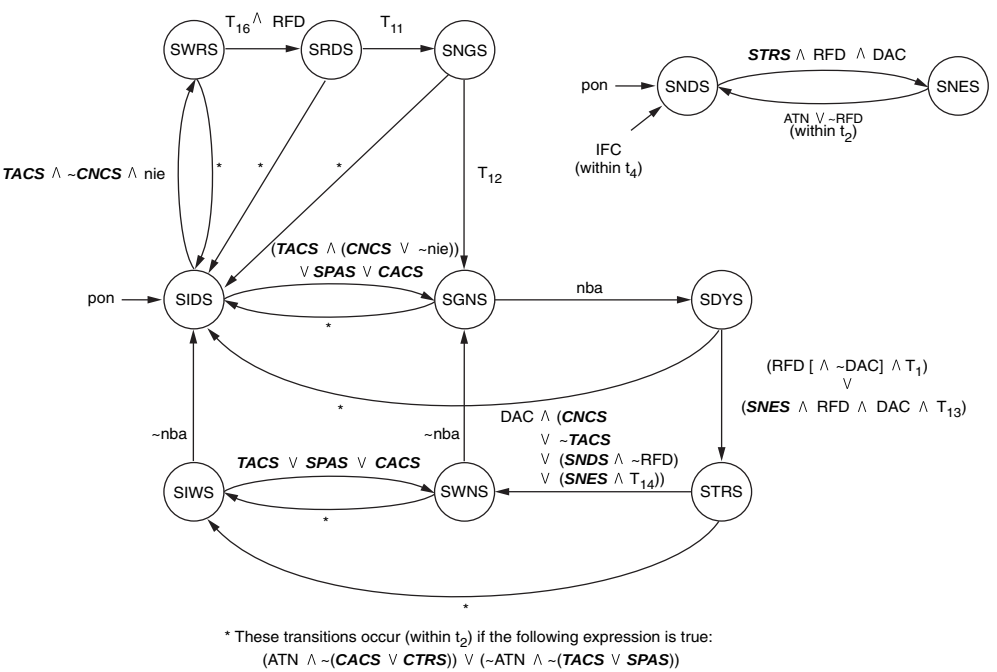


Figure 4—SHE state diagram

Table 2—SH mnemonics

Messages		Interface States	
Mnemonic	Definition	Mnemonic	Definition
pon	power on	SIDS	source idle state
nba	new byte available	SGNS	source generate state
ATN	attention	SDYS	source delay state
RFD	ready for data	STRS	source transfer state
DAC	data accepted	SWNS	source wait for new cycle state
		SIWS	source idle wait state
		TACS	talker active state (T function)
		SPAS	serial poll active state (T function)
		CACS	controller active state (C function)
		CTRS	controller transfer state (C function)

Table 3—SHE mnemonics

Messages		Interface States	
Mnemonic	Definition	Mnemonic	Definition
pon	power on	SIDS	source idle state
nba	new byte available	SGNS	source generate state
nie	noninterlocked enable	SDYS	source delay state
		STRS	source transfer state
IFC	interface clear	SWNS	source wait for new cycle state
ATN	attention	SIWS	source idle wait state
RFD	ready for data	SNDS	source noninterlocked disable state
DAC	data accepted	SNES	source noninterlocked enable state
		SWRS	source wait for RFD state
		SRDS	source RFD delay state
		CNCS	configure not configured state (CF function)
		TACS	talker active state (T function)
		SPAS	serial poll active state (T function)
		CACS	controller active state (C function)
		CTRS	controller transfer state (C function)

Table 4—SH message outputs

SH State	Remote Message Sent	Device Function (DF) Interaction
	DAV	
SIDS	(F)	DF can change remote multiline messages
SGNS	F	DF can change remote multiline messages
SDYS	F	DAB, EOS multiline, and END messages shall not change
STRS	T	DAB, EOS multiline, and END messages shall not change
SWNS	T or F	DF requested to change multiline messages
SIWS	(F)	DF requested to change multiline messages

Table 5—SHE message outputs

SHE State	Remote Message Sent		Device Function (DF) Interaction
	DAV	NIC	
SIDS	(F)	(F)	DF can change remote multiline messages
SGNS	F	(F)	DF can change remote multiline messages
SDYS	F	(F)	DAB, EOS multiline, and END messages shall not change
STRS	T	(F)	DAB, EOS multiline, and END messages shall not change
SWNS	T or F	(F)	DF requested to change multiline messages
SIWS	(F)	(F)	DF requested to change multiline messages
SWRS	F	(F)	DF can change remote multiline messages
SRDS	F	(F)	DF can change remote multiline messages
SNGS	F	T	DF can change remote multiline messages

4.3.3 SH function state descriptions

4.3.3.1 Source idle state (SIDS)

In SIDS, the SH function, or the SHE function, is not engaged in the handshake cycle and does not have a new message byte available. The SH function, or the SHE function, powers on in SIDS.

In SIDS, the SH function shall send the DAV message passively false. In SIDS, the SHE function shall send the DAV and noninterlocked capable (NIC) messages passively false.

The SH function shall exit SIDS and enter the source generate state (SGNS) if

- a) The talker active state (TACS) is active
- b) Or the serial poll active state (SPAS) is active
- c) Or the controller active state (CACS) is active

The SHE function shall exit SIDS and enter

- a) The SGNS if either
 - 1) The SPAS is active
 - 2) Or the CACS is active
 - 3) Or the TACS is active and either the CNCS is active or the nie message is false
- b) The SWRS if the TACS is active and the CNCS is not active and the nie message is true

4.3.3.2 Source generate state (SGNS)

In SGNS, the device is generating a new message byte and the function is waiting for the new byte to become available.

In SGNS, the SH function shall send the DAV message false. In SGNS, the SHE function shall send the DAV message false and send the noninterlocked capable (NIC) message passively false. In this state, the device may change the multiline message being sent via the talker or controller interface function while in TACS or CACS or SPAS.

The SH function, or the SHE function, shall exit SGNS and enter

- a) The source delay state (SDYS) if the new byte available (nba) message is true
- b) The SIDS within t_2 if either
 - 1) The ATN message is true and neither CACS nor CTRS is active
 - 2) Or the ATN message is false and neither TACS nor SPAS is active

4.3.3.3 Source delay state (SDYS)

In SDYS, the SH function, or the SHE function, is waiting for a message byte to settle on the interface signal lines after the change during SGNS. In SDYS the SH function, or the SHE function (if it is using interlocked handshaking), is also waiting for all of the acceptor functions to indicate their readiness to accept the message byte.

In SDYS, the SH function shall send the DAV message false. In SDYS, the SHE function shall send the DAV message false and send the NIC messages passively false. In this state, the device shall not change the multiline message being sent.

The SH function shall exit SDYS and enter

- a) The source transfer state (STRS) only after T_1 , if the RFD message is true and if optionally, the DAC message is false
- b) The SIDS within t_2 if either
 - 1) The ATN message is true and neither CACS nor CTRS is active
 - 2) Or the ATN message is false and neither TACS nor SPAS is active

The SHE function shall exit SDYS and enter

- a) The STRS if either
 - 1) The RFD message is true, only after T_1 , and if optionally, the DAC message is false
 - 2) Or the SNES is active and the RFD and DAC messages are true (only after T_{13})
- b) The SIDS within t_2 if either
 - 1) The ATN message is true and neither CACS nor CTRS is active
 - 2) Or the ATN message is false and neither TACS nor SPAS is active

4.3.3.4 Source transfer state (STRS)

In STRS, the SH function, or the SHE function, indicates to the AH function, or AHE function, that it is continuously sending a valid message byte.

In STRS, the SH function shall send the DAV message true. In STRS, the SHE function shall send the DAV message true and send the NIC messages passively false. In this state, the device shall not change either the multiline message or the END message (if used) being sent.

The SH function shall exit STRS and enter

- a) The source idle wait state (SIWS) within t_2 if either
 - 1) The ATN message is true and neither CACS nor CTRS is active
 - 2) Or the ATN message is false and neither TACS nor SPAS is active
- b) The source wait for new cycle state (SWNS) if the DAC message is true

The SHE function shall exit STRS and enter

- a) The source idle wait state (SIWS) within t_2 if either
 - 1) The ATN message is true and neither CACS nor CTRS is active
 - 2) Or the ATN message is false and neither TACS nor SPAS is active
- b) The SWNS if the DAC message is true and either
 - 1) The CNCS is active
 - 2) Or the TACS is not active
 - 3) Or the SNDS is active and RFD message is false
 - 4) Or the SNES is active after T_{14}

4.3.3.5 Source wait for new cycle state (SWNS)

In SWNS, the SH function, or the SHE function, is waiting for the device to start a new message generation cycle.

In SWNS, the SH function shall send the DAV message true or false. In SWNS, the SHE function shall send the DAV true or false and the NIC messages passively false.

The SH function, or the SHE function, shall exit SWNS and enter

- a) The SGNS if the nba message is false
- b) The SIWS within t_2 if either
 - 1) The ATN message is true and neither CACS nor CTRS is active
 - 2) Or the ATN message is false and neither TACS nor SPAS is active

4.3.3.6 Source idle wait state (SIWS)

In SIWS, the SH function, or the SHE function, is not active in the external message byte transfer process but is active in the internal process of waiting for the device to start a new message generation cycle. This SIWS allows a sequence of message byte transfers to be interrupted without loss of data over the interface while the device may continue to prepare for the new (next) message byte generation cycle.

In SIWS, the SH function shall send the DAV message passively false. In SIWS, the SHE function shall send the DAV and NIC messages passively false.

The SH function, or the SHE function, shall exit SIWS and enter

- a) The SIDS if the nba message is false
- b) The SWNS if either
 - 1) The TACS is active
 - 2) Or the SPAS is active
 - 3) Or the CACS is active

4.3.3.7 Source wait for RFD state (SWRS)

In SWRS, the SHE function is waiting for all acceptor functions to indicate their readiness to accept the first DAB since the most recent false transition of ATN.

NOTE—The SHE will enter SWRS to initiate a noninterlocked mode of data transfer. SWRS can only be entered if CNCS is false. CNCS can only be false if the controller explicitly issues a CFGn command. It is a REQUIREMENT that all noninterlocked handshake mode features default (power-on) to disabled until an explicit CFGn command is issued.

In SWRS, the SHE function shall send the DAV false and send the NIC messages passively false.

The SHE function shall exit SWRS and enter

- a) The SRDS if the RFD message is true (only after T_{16})
- b) Or the SIDS within t_2 if either
 - 1) The ATN message is true and neither CACS nor CTRS is active
 - 2) Or the ATN message is false and neither TACS nor SPAS is active

4.3.3.8 Source RFD delay state (SRDS)

In SRDS, the SHE is waiting for all acceptors to see the RFD message true before issuing the NIC message. All acceptors must observe the RFD message true before the NIC message is issued to distinguish between the RFD message of a slower acceptor and the source's NIC message.

In SRDS, the SHE function shall send the DAV message false and the NIC message passively false.

The SHE function shall exit SRDS and enter either

- a) The SNGS only after T_{11}
- b) Or the SIDS within t_2 if either
 - 1) The ATN message is true and neither CACS nor CTRS is active
 - 2) Or the ATN message is false and neither TACS nor SPAS is active

4.3.3.9 Source NIC generate state (SNGS)

In SNGS, the SHE function indicates to all acceptor functions that it is capable of sourcing bytes using non-interlocked handshake cycles.

In SNGS, the SHE function shall send the DAV message false and the NIC message true.

The SHE function shall exit SNGS and enter

- a) The SGNS only after T_{12}
- b) The SIDS within t_2 if either
 - 1) The ATN message is true and neither CACS nor CTRS is active
 - 2) Or the ATN message is false and neither TACS nor SPAS is active

4.3.3.10 Source noninterlocked disable state (SNDS).

In SNDS, the SHE function is not capable of sourcing multiline message bytes using noninterlocked handshake cycles. The SHE function powers on in SNDS.

The SHE function shall exit SNDS and enter SNES if

- a) The STRS is active
- b) And the DAC message is true
- c) And the RFD message is true
- d) And the IFC message is false

4.3.3.11 Source noninterlocked enable state (SNES)

In SNES, the SHE function is capable of sourcing multiline message byte using noninterlocked handshake cycles.

The SHE function shall exit SNES and enter SNDS if either

- a) the ATN message is true (within t_2)
- b) Or the RFD message is false (within t_2)
- c) Or the IFC message is true (within t_4)

4.3.4 SH function and SHE function allowable subsets

The only allowable subsets to the SH and SHE interface functions shall be those listed in Tables 6 and 7.

Table 6—Allowable subsets to SH function

Identification	Description	States Omitted	Other Requirements	Other Function Subsets Required
SH0	no capability	all	none	none
SH1	complete capability	none	none	T1-T8, TE1-TE8, or C5-C28

Table 7—Allowable subsets to SHE function

Identification	Description	States Omitted	Other Requirements	Other Function Subsets Required
SHE0	no capability	all	none	none
SHE1	complete capability	none	none	CF1 and T1-T8, TE1-TE8, or C5-C28

4.3.5 Additional SH and SHE function requirements and guidelines

The nba true message indicates the device has generated a (new) message byte and made it available on the interface signal lines.

The nba message shall become true only in SIDS, SWRS, SRDS, SNGS, or SGNS. The nba message may become false in any other SH or SHE states.

The expression for interruption ($ATN \wedge \overline{CACS} \vee \overline{CTRS} \vee (\overline{ATN} \wedge \overline{TACS} \vee \overline{SPAS})$) may be substituted by $\overline{TACS} \wedge \overline{SPAS} \wedge \overline{CACS} \wedge \overline{CTRS}$ if the transition of the latter expression can be effected within t_2 after the change of ATN.

4.4 Acceptor handshake (AH) and extended acceptor handshake (AHE) interface functions

4.4.1 General description

The AH function provides a device with the capability to guarantee proper reception of remote multiline messages. Two alternative versions of the function exist: the AH function and the extended acceptor handshake (AHE) function. The AHE interface function is a superset of the AH interface function. Only one of the two alternative functions shall be implemented in a specific device.

The AH function may delay either the initiation of, or termination of, a multiline message transfer until prepared to continue with the transfer process. The AH function utilizes the DAV, RFD, and DAC messages to effect each message byte transfer. When a SHE function is transferring data to one or more AHE functions using noninterlocked handshake cycles, an AHE function may either delay the initiation of multiline message bytes or force the SHE to use interlocked handshake cycles.

Transfers between an SHE function and an AHE function may use noninterlocked handshake cycles. Other transfers use interlocked handshake cycles.

NOTE—Both the AH function and the AHE function are described concurrently throughout 4.4 due to the extensive similarity between these two functions.

4.4.2 AH function state diagram

The AH interface function shall be implemented so as to perform according to the state diagram given in Figure 5 and the state descriptions given throughout 4.4. Table 8 specifies the set of messages and states required to effect transition from one active state to another. Table 10 specifies the messages that shall be sent and the device function interaction required while each state is active.

The AHE interface function shall be implemented so as to perform according to the state diagram given in Figure 6 and the state descriptions given throughout 4.4. Table 9 specifies the set of messages and states required to effect transition from one active state to another. Table 11 specifies the messages that shall be sent and the device function interaction required while each state is active.

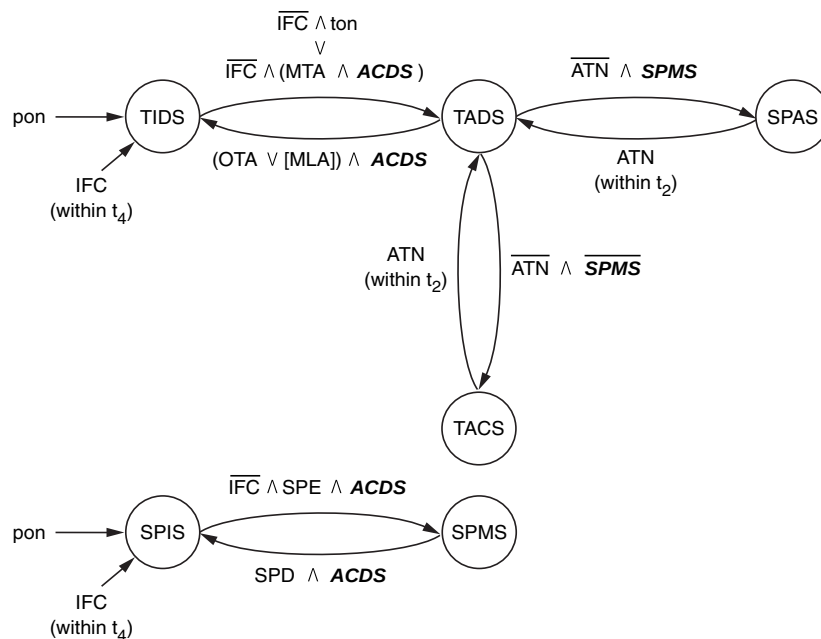


Figure 5—AH state diagram

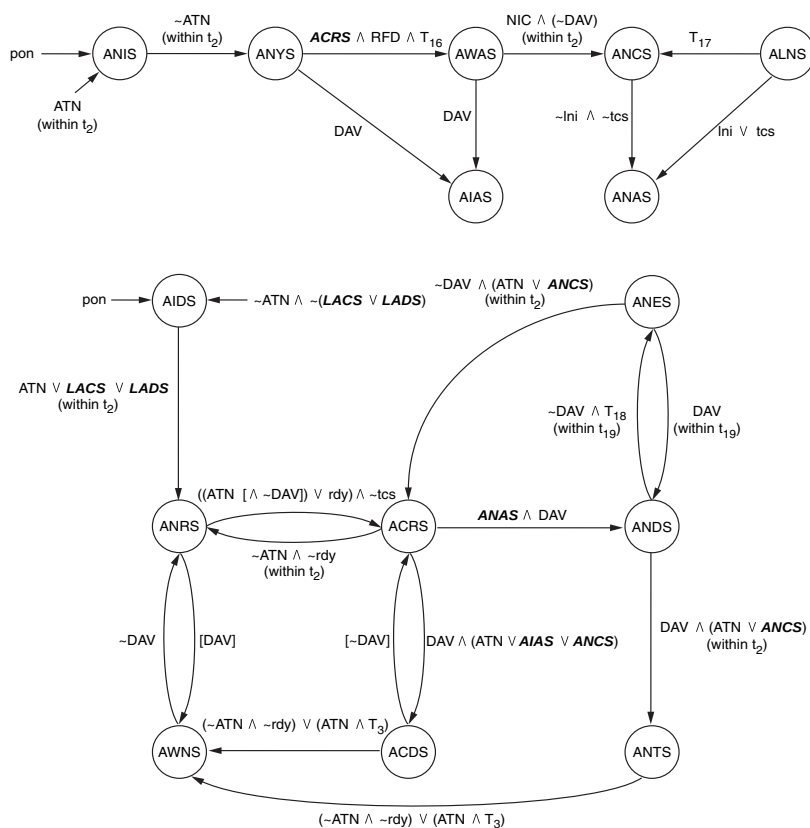


Figure 6—AH state diagram

Table 8—AH mnemonics

Messages		Interface States	
Mnemonic	Definition	Mnemonic	Definition
pon	power on	AIDS	acceptor idle state
rdy	ready for next message	ANRS	acceptor not ready state
tcs	take control synchronously ^a	ACRS	accept ready state
ATN	attention	ACDS	accept data state
DAV	data valid	AWNS	acceptor wait for new cycle state
		LADS	listener addressed state (L function)
		LACS	listener active state (L function)

^aSee the first paragraph of 4.12.3.7.

Table 9—AHE mnemonics

Messages		Interface States	
Mnemonic	Definition	Mnemonic	Definition
nba	new byte available	AIDS	acceptor idle state
pon	power on	ANRS	acceptor not ready state
rdy	ready for next message	ACRS	accept ready state
tcs	take control synchronously	ACDS	accept data state
lni	leave noninterlocked	AWNS	acceptor wait for new cycle state
rft	ready for three	ANDS	accept noninterlocked ready state
		ANES	accept noninterlocked not ready state
ATN	attention	ANTS	accept noninterlocked terminate state
DAV	data valid	ANIS	accept noninterlocked inactive state
RFD	Ready For Data	ANYS	accept noninterlocked delay state
NIC	noninterlocked Capable	AWAS	accept wait for noninterlocked capable state
		AIAS	accept interlocked always state
		ANCS	accept noninterlocked configured state
		ANAS	accept noninterlocked active state

Table 9—AHE mnemonics (continued)

Messages		Interface States	
Mnemonic	Definition	Mnemonic	Definition
		ALNS	accept leave noninterlocked state
		LADS	listener addressed state (L function)
		LACS	listener active state (L function)

Table 10—AH message outputs

AH State	Remote Message Sent		Device Function (DF) Interaction
	RFD	DAC	
AIDS	(T)	(T)	DF cannot receive multiline or END messages
ANRS	F	F	DF cannot receive multiline or END messages
ACRS	(T)	F	DF cannot receive multiline or END messages
ACDS	F	F	DF can receive multiline or END messages if LACS is active
AWNS	F	(T)	DF cannot receive multiline or END messages

Table 11—AHE message outputs

AHE State	Qualifier	Remote Message Sent		Device Function (DF) Interaction
		RFD	DAC	
AIDS		(T)	(T)	DF cannot receive multiline or END messages
ANRS		F	F	DF cannot receive multiline or END messages
ACRS		(T)	F	DF cannot receive multiline or END messages
AWNS		F	(T)	DF cannot receive multiline or END messages
ACDS		F	F	DF can receive multiline or END messages if LACS is active
ANDS	ANAS $\wedge$ rft	(T)	(T)	DF can receive multiline or END messages (using noninterlocked handshaking) if LACS is active
ANDS	(ANAS $\wedge$ rft)	(T)	F	DF can receive multiline or END messages (using noninterlocked handshaking) if LACS is active

Table 11—AHE message outputs (*continued*)

AHE State	Qualifier	Remote Message Sent		Device Function (DF) Interaction
		RFD	DAC	
ANES	ANAS $\wedge$ rft	(T)	(T)	DF cannot receive multiline or END messages
ANES	(ANAS $\wedge$ rft)	(T)	F	DF cannot receive multiline or END messages
ANTS		F	F	DF can receive multiline or END messages (using noninterlocked handshaking) if LACS is active

4.4.3 AH function state descriptions

4.4.3.1 Acceptor idle state (AIDS)

In AIDS, the AH function, or AHE function, is inactive and not engaged in the handshake cycle. The AH function, or AHE function, powers on in AIDS.

In AIDS, the RFD and DAC messages shall be sent passive true.

The AH function, or AHE function, shall exit AIDS and enter the acceptor not ready state (ANRS) within t_2 if either

- a) The ATN message is true
- b) Or LACS is active
- c) Or LADS is active

4.4.3.2 Acceptor not ready state (ANRS)

In ANRS, the AH function, or AHE function, indicates to the interface it has not yet prepared internally to continue with the handshake cycle.

In ANRS, the RFD and DAC messages shall be sent false.

The AH function, or AHE function, shall exit ANRS and enter

- a) The ACRS if the take control synchronously (tcs) message is false (see the first paragraph of 4.12.3.7) and either
 - 1) The ATN message is true and the DAV message is false
 - 2) Or the ready for next message (rdy) message is true

NOTE—Use of the DAV message is optional.

- b) The AIDS if the ATN message is false and neither
 - 1) The LADS is active
 - 2) Nor LACS is active
- c) The Awns if, optionally, the DAV message is true (note that this transition will never occur under normal interface operation)

4.4.3.3 Acceptor ready state (ACRS)

In ACRS, the AH function, or AHE function, indicates to the interface that it is prepared to receive multiline messages using interlocked handshaking.

In ACRS, the DAC message shall be sent false and the RFD message shall be sent passive true.

The AH function shall exit ACRS and enter

- a) The accept data state (ACDS) if the DAV message is true
- b) The AIDS if the ATN message is false and neither
 - 1) The LADS is active
 - 2) Nor LACS is active
- c) The ANRS within t_2 if both the ATN and the rdy message are false

The AHE function shall exit ACRS and enter

- a) The accept data state (ACDS) if the DAV message is true and either
 - 1) The ATN message is true
 - 2) Or the AIAS is active
 - 3) Or the ANCS is active
- b) The ANDS if the DAV message is true and the ANAS is active
- c) The AIDS if the ATN message is false and neither
 - 1) The LADS is active
 - 2) Nor LACS is active
- d) The ANRS within t_2 if both the ATN and the rdy messages are false

4.4.3.4 Accept data state (ACDS)

In ACDS, the AH function, or AHE function, indicates to the SH function that it shall maintain a valid message byte. In this state, the multiline messages on the DIO signal lines are valid. The ACDS indicates to the interface functions that an interface message is present and valid if the ATN message is true. The ACDS indicates to the device functions that a device-dependent message is present and valid if LACS is active.

In ACDS, the DAC and RFD messages shall be sent false.

The AH function, or AHE function, shall exit the ACDS and messages enter

- a) The acceptor wait for new cycle state (AWNS) if either
 - 1) The ATN message is true and a period of T_3 has elapsed
 - 2) Or the ATN and rdy messages are both false
- b) The AIDS if the ATN message is false and neither
 - 1) The LADS is active
 - 2) Nor LACS is active
- c) The ACRS if, optionally, the DAV message is false (note that this transition can occur only when the controller takes control asynchronously)

4.4.3.5 Acceptor wait for new cycle state (AWNS)

In AWNS, the AH function, or AHE function, indicates that it has received a multiline message byte.

In AWNS, the RFD message shall be sent false and the DAC message shall be sent passive true.

The AH function shall exit the AWNS and enter

- a) The ANRS if DAV is false
- b) The AIDS if the ATN message is false and neither
 - 1) The LADS is active
 - 2) Nor LACS is active

4.4.3.6 Accept noninterlocked ready state (ANDS)

In ANDS, the AHE function has accepted a data byte using noninterlocked handshaking. The AHE function shall accept the data byte upon entry to ANDS.

In ANDS, the RFD messages shall be sent passively true. The DAC message shall be sent passively true if the ANAS is active and the ready for three (rft) local message is true. The DAC message shall be sent false if the ANAS is inactive or the rft local message is false.

The ready for three bytes (rft) local message indicates the buffer the device uses to accept incoming multi-line message bytes has room for at least three more bytes and must become false before entry into ANDS that accepts the byte, making the device two from full. The rft local message can be used to stop the transfer for other reasons, which may be asynchronous to the transfer; however, in this case, more than three bytes may be received before the transfer is stopped.

The AHE function shall exit ANDS and enter

- a) The AIDS if the ATN message is false and neither
 - 1) The LADS is active
 - 2) Nor LACS is active
- b) The ANES, after T_{18} but within t_{19} , if the DAV message is false
- c) The ANTS within t_2 if the DAV message is true and either the ATN message is true or the ANCS is active

4.4.3.7 Accept noninterlocked not ready state (ANES)

In ANES, the AHE function is prepared to receive multiline messages using noninterlocked handshaking.

In ANES, the RFD messages shall be sent passively true. The DAC message shall be sent passively true if the ANAS is active and the ready for three (rft) local message is true. The DAC message shall be sent false if the ANAS is inactive or if the rft local message is false.

The AHE function shall exit ANES and enter

- a) The AIDS if the ATN message is false and neither
 - 1) The LADS is active
 - 2) Nor LACS is active
- b) The ANDS within t_{19} if the DAV message is true
- c) The ACRS within t_2 if the DAV message is false and either the ATN message is true or the ANCS is active

4.4.3.8 Accept noninterlocked terminate state (ANTS)

In ANTS, the AHE function indicates that it is resuming interlocked handshaking. In ANTS, the RFD and DAC messages shall be sent false.

The AHE function shall exit ANTS and enter

- a) The AIDS if the ATN message is false and neither
 - 1) The LADS is active
 - 2) Nor LACS is active
- b) The AWNS if either
 - 1) The ATN message is true and a period of T_3 has elapsed
 - 2) Or the ATN and rdy messages are both false

4.4.3.9 Accept noninterlocked inactive state (ANIS)

In ANIS, the AHE function is not capable of using noninterlocked handshaking. The AHE function powers on in ANIS.

The AHE function shall exit ANIS and enter ANYS (within t_2) if the ATN message is false.

4.4.3.10 Accept noninterlocked delay state (ANYS)

In ANYS, the AHE function is waiting for all acceptors to enter either ACRS or AIDS after a true to false transition of ATN. All acceptors must have ACRS or AIDS active before the sourcing device may send the NIC message true.

The AHE function shall exit ANYS and enter either

- a) The ANIS, within t_2 , if the ATN message is true
- b) Or the AWAS, after T_{16} , if the ACRS is active and the RFD message is true
- c) Or the AIAS if the DAV message is true

4.4.3.11 Accept wait for noninterlocked capable state (AWAS)

In AWAS, the AHE function is waiting for either:

- a) The sourcing device to send the DAV message true—indicating that the sourcing device will transmit multiline messages using interlocked handshake cycles.
- b) The sourcing device to send the NIC message true—indicating that the sourcing device is capable of sending multiline messages using noninterlocked handshake cycles.

The AHE function shall exit AWAS and enter either

- a) The ANIS, within t_2 , if the ATN message is true
- b) Or the AIAS if the DAV message is true
- c) Or the ANCS if the NIC message is true and the DAV message is false

4.4.3.12 Accept interlocked always state (AIAS)

In AIAS, the AHE function has detected that the sourcing device has sent the first data byte without sending the NIC message. In AIAS, the AHE function will accept data bytes using the interlocked handshake.

The AHE function shall exit AIAS and enter the ANIS, within t_2 , if the ATN message is true.

4.4.3.13 Accept noninterlocked configured state (ANCS)

In ANCS, the AHE function has detected that the sourcing device has sent the NIC message, but the AHE function is not ready to accept data bytes using noninterlocked handshake cycles.

If the leave noninterlocked (lni) local message is true, the device will always perform interlock handshaking on all multiline message bytes. If lni local message is false and the source issues the NIC message, the device will perform noninterlock handshaking on all multiline message bytes.

NOTE—The AHE will enter ANCS to initiate a noninterlocked mode of data transfer. ANCS can only be entered if NIC is true, indicating the SHE is in SNGS. The SHE can only be entered if previously CNCS was false. CNCS can only be false if the controller explicitly issues a CFGn command. It is a REQUIREMENT that all noninterlocked handshake mode features default (power-on) to disabled until an explicit CFGn command is issued.

The AHE function shall exit ANCS and enter either

- a) The ANIS, within t_2 , if the ATN message is true
- b) Or the ANAS if the leave noninterlocked (lni) local message is false and the take control synchronously (tcs) local message is false

4.4.3.14 Accept noninterlocked active state (ANAS)

In ANAS, the AHE function is capable of accepting data bytes using noninterlocked handshaking.

The AHE function shall exit ANAS and enter either

- a) The ANIS, within t_2 , if the ATN message is true
- b) Or the ALNS if the lni local message is true or the tcs local message is true

4.4.3.15 Accept leave noninterlocked state (ALNS)

In ALNS, the AHE function is preparing to stop accepting data bytes using noninterlocked handshake cycles. By leaving ANAS, the DAC message will be sent false while the ANES or ANDS are active (see 4.4.3.6 and 4.4.3.7). In ALNS, the AHE function is waiting for the SH function to detect the DAC message false and stop sending multiline message bytes.

The AHE function shall exit ALNS and enter either

- a) The ANIS, within t_2 , if the ATN message is true
- b) Or the ANCS after T_{17}

4.4.4 AH function and AHE function allowable subsets

The only allowable subsets to the AH and AHE interface functions shall be those listed in Tables 12 and 13, respectively.

4.4.5 Additional AH and AHE function requirements and guidelines

The local message rdy shall not become false during the ACRS. The transition from ACRS to ANRS shall occur only at the time ATN becomes false.

The RFD message received by an SH function is the logical AND of all the RFD messages sent by all the active AH functions. Similarly, the DAC message received by an SH function is the logical AND of all the DAC messages sent by all the AH functions. The way in which the composite effects of multiple AH functions interact with an SH function to perform the logical AND function via the use of the NRFD and NDAC signal lines is explained further in 7.4.

As interface functions need be designed only so as to perform according to the state diagrams specified, it is not required that exactly the states specified are the ones that exist in an implementation. One consequence

Table 12—Allowable subsets to AH function

Identification	Description	States Omitted	Other Requirements	Other Function Subsets Required
AH0	no capability	all	none	none
AH1	complete capability	none	none	none

Table 13—Allowable subsets to AHE function

Identification	Description	States Omitted	Other Requirements	Other Function Subsets Required
AHE0	no capability	all	none	none
AHE1	complete capability	none	none	CF1

of this statement is that interface function state transitions that are qualified by interface messages can occur after the message has been received as long as the RFD message is held false until they occur. The resulting performance cannot be distinguished from the performance of the specified diagrams in which the transitions shall occur while the interface message is being received. If this type of implementation is chosen, then the AH function should remain in ANRS even though the exit condition is true in order to hold the RFD message false (this is allowed by the last paragraph in 4.1.3.2).

In noisy environments, filter action on the incoming DAV message within a device can minimize false transitions to the ACDS state.

The AHE function may resume interlocked handshaking at any point during a transfer. A noninterlock-capable listener may choose not to use noninterlocked handshake cycles for any reason. For example, if a noninterlocked-capable listener is only planning to accept a few bytes, it may accept them all using interlocked handshake cycles.

The AHE function requires a buffer of at least 3 bytes for receiving multiline messages.

If a device makes the tcs message true to cause a transition from the CSBS to the CSHS, the device normally should also make the local lni message true. lni forces the AHE function to return to interlocked handshaking, which is required for taking control synchronously.

4.5 Talker (T) interface function (Includes serial poll capabilities)

4.5.1 General description

The T interface function provides a device with the capability to send device-dependent data (including status data during a serial poll sequence) over the interface to other devices. This capability exists only when the T interface function is addressed to talk.

There are two alternative versions of the function: one with and one without address extension. The normal T function uses a 1 byte address, the primary talk address. The T interface function with address extension [hereinafter called a TE (extended talker) function] uses a 2 byte address, the primary and secondary talk addresses. In all other respects, the capabilities of both versions are the same.

Only one of the two alternative T functions needs to be implemented in a specific device.

NOTE—Both the T function and the TE function are described concurrently throughout 4.5 due to the extensive similarity between these two functions.

4.5.2 T function state diagrams

The T function shall be implemented so as to perform according to the state diagrams given in Figure 7 and the state descriptions given throughout 4.5. Table 14 specifies the set of messages and states required to effect transition from one active state to another. Table 15 specifies the messages that shall be sent and the device function interaction required while each state is active.

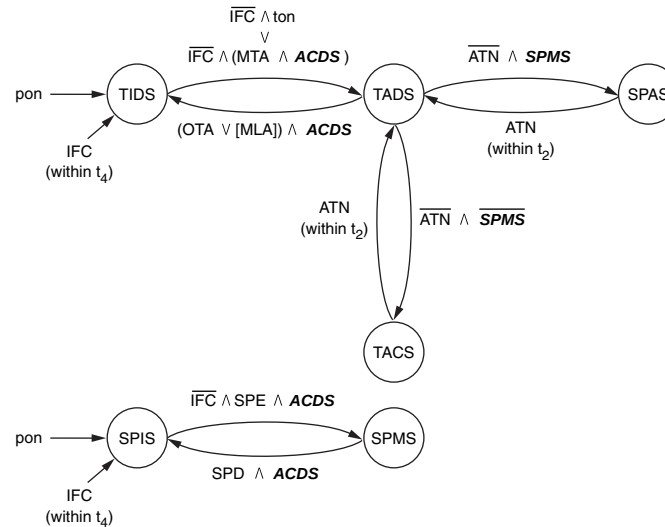


Figure 7—T state diagram

The TE function shall be implemented so as to perform according to the state diagrams given in Figure 8 and the state descriptions given throughout 4.5. Table 16 specifies the set of messages and states required to effect transition from one active state to another. Table 15 specifies the messages that shall be sent and the device function interaction required while each state is active.

Table 14—T mnemonics

Messages		Interface States	
Mnemonic	Definition	Mnemonic	Definition
pon	power on	TIDS	talker idle state
ton	talk only	TADS	talker addressed state
IFC	interface clear	TACS	talker active state
ATN	attention	SPAS	serial poll active state
MTA	my talk address	SPIS	serial poll idle state
SPE	serial poll enable	SPMS	serial poll mode state
SPD	serial poll disable	ACDS	accept data state (AH function)
OTA	other talk address		
MLA	my listen address		

Table 15—T or TE message outputs

T State	Qualifier	Remote Messages Sent ^a			Device Function (DF) Interaction
		Multiline	END	RQS ^b	
TIDS		(NUL)	(F)	(F)	DF not allowed to send messages
TADS		(NUL)	(F)	(F)	DF not allowed to send messages
TACS		DAB ^c or EOS ^c	T or F ^c	(F)	DF can send DAB, EOS, or END message (if used) concurrent with DAB ^d
SPAS	APRS inactive	STB ^c	F or T	F	DF can send one STB message ^d
SPAS	APRS active	STB ^c	F or T	T	DF can send one STB message ^d

^aSee Table 44, 4.13.

^bSee 4.5.3.4.

^cMessages enabled by the T function originating within the device functions.

^dUnder SH control.

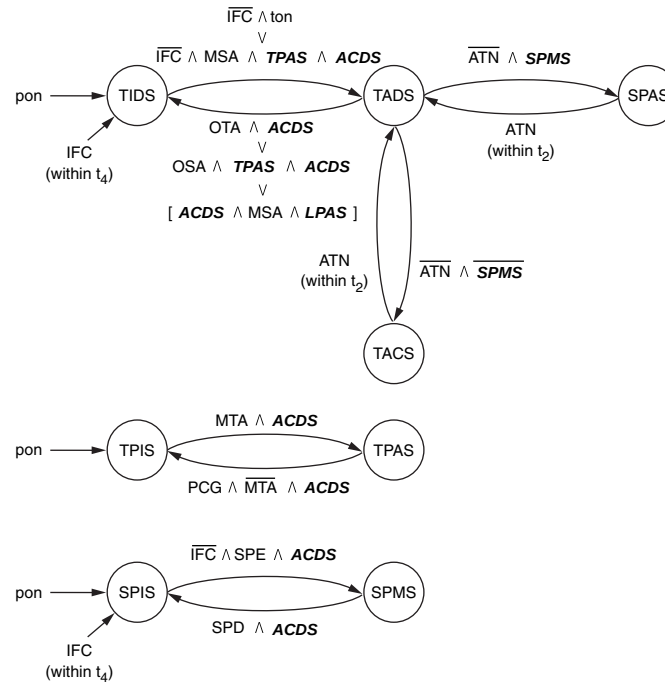


Figure 8—TE state diagram

Table 16—TE mnemonics

Messages		Interface States	
Mnemonic	Definition	Mnemonic	Definition
pon	power on	TIDS	talker idle state
ton	talk only	TADS	talker addressed state
IFC	interface clear	TACS	talker active state
ATN	attention	SPAS	serial poll active state
MTA	my talk address	TPIS	talker primary idle state
OTA	other talk address	TPAS	talker primary addressed state
OSA	other secondary address	SPIS	serial poll idle state
PCG	primary command group	SPMS	serial poll mode state
SPE	serial poll enable	ACDS	accept data state (AH function)
SPD	serial poll disable	LPAS	listener primary addressed state (L function)
MSA	my secondary address		

4.5.3 T function state descriptions

4.5.3.1 Talker idle state (TIDS)

In TIDS, neither the T function nor the TE function is engaged in sending data or status bytes. The T function or the TE function powers on in TIDS.

In TIDS, the END and request service (RQS) messages shall be sent passive false and the NUL message shall be sent passive true.

The T function shall exit the TIDS when the IFC message is false and enter the talker addressed state (TADS) if either

- a) The my talk address (MTA) message is true and ACDS is active
- b) Or the talk only (ton) message is true (see the last paragraph of 4.5.5)

The TE function shall exit TIDS and enter TADS if the IFC message is false and either

- a) The my secondary address (MSA) message is true, and ACDS is active, and the talker primary address state (TPAS) is active
- b) Or the ton message is true

4.5.3.2 Talker addressed state (TADS)

In TADS, the T function has received its talk address and is prepared for, but not engaged in, sending data or status bytes. In TADS, the TE function has received both its primary and secondary talk addresses and is prepared for, but not engaged in, sending data or status bytes.

In TADS, the END and RQS messages shall be sent passive false and the NUL message shall be sent passive true.

The T function shall exit TADS and enter

- a) The talker active state (TACS) if the ATN message is false and the serial poll mode state (SPMS) is inactive
- b) The serial poll active state (SPAS) if the ATN message is false and SPMS is active
- c) The TIDS if either
 - 1) The other talk address (OTA) message is true and ACDS is active
 - 2) Or the MLA message is true and ACDS is active
 - 3) Or within t_4 if the IFC message is true

NOTE—Use of the expression containing the MLA message is optional.

The TE function shall exit TADS and enter

- a) The TACS if the ATN message is false and SPMS is inactive
- b) The SPAS if the ATN message is false and SPMS is active
- c) The TIDS if either
 - 1) The OTA message is true and ACDS is active
 - 2) Or the other secondary address (OSA) message is true and TPAS and ACDS are active
 - 3) Or the MSA message is true and both the listener primary addressed state (LPAS) and ACDS are active
 - 4) Or within t_4 if the IFC message is true

NOTE—Use of the expression containing the MSA message is optional.

4.5.3.3 Talker active state (TACS)

In TACS, the T function, or the TE function, enables the transfer of the data byte (DAB) message and END, if used, from the device function to the interface signal lines. The message content is determined solely by the device function(s). The SH function determines when the device function(s) may change the message content of DAB (and END if used).

During TACS, the DAB or end of string (EOS) and END messages may be sent by the device functions. The RQS message shall be sent passive false.

NOTE—The coding and format of the data are, in general, device-dependent and beyond the scope of this standard.

The T function or the TE function shall exit TACS and enter

- a) The TADS within t_2 if the ATN message is true
- b) The TIDS within t_4 if the IFC message is true

4.5.3.4 Serial poll active state (SPAS)

In SPAS, the T function or the TE function enables the transfer of a single status message from the device function to the interface signal lines using the SH or SHE function to control the transfer of the status byte that contains both the RQS and the STB messages.

Although a controller needs only 1 byte for the STB and RQS messages from a device, it is allowable for the device to repeat this combined message byte if the controller does not assert ATN after the first transfer. In this case, the content of the STB message may change between subsequent transfers although the RQS message is held unaltered by the SR function.

During SPAS, whether APRS state is active or inactive, the END message shall be sent either true or false. The RQS message shall be sent true if APRS is active, or false if APRS is inactive. In addition, the STB message shall be sent by the device function(s).

NOTE—The APRS is contained in the SR interface function.

The T function or the TE function shall exit SPAS and enter

- a) The TADS within t_2 if the ATN message is true
- b) The TIDS within t_4 if the IFC message is true

4.5.3.5 Serial poll idle state (SPIS)

In SPIS, the T function or the TE function is not enabled to participate in a serial poll. The T or TE function powers on in SPIS.

The SPIS does not provide a remote message sending capability.

The T function or the TE function shall exit SPIS and enter SPMS if the serial poll enable (SPE) message is true and ACDS is active and the IFC message is false.

4.5.3.6 Serial poll mode state (SPMS)

In SPMS, the T function or the TE function is enabled to participate in a serial poll.

The SPMS does not provide a remote message sending capability.