

JIS

JAPANESE INDUSTRIAL STANDARD

**An Interface System for Program-
mable Measuring Instruments**

JIS C 1901—1987

Translated and Published

by

Japanese Standards Association

Printed in Japan

In the event of any doubt arising,
the original Standard in Japanese is to be final authority.

CONTENTS

Section 1. General	1
1. Scope and Object	1
1.1 Scope	1
1.2 Object	2
2. Definitions	2
2.1 General System Terms	3
2.2 Signals and Paths	4
3. Interface System Overview	4
3.1 Interface System Objective	4
3.2 Fundamental Communication Capabilities	5
3.3 Message Paths and Bus Structure	6
3.4 Interface System Elements	8
Section 2. Functional Specifications	9
4. Functional Partition	9
4.1 Device Functions	9
4.2 Interface Function Concepts	9
4.3 Message Concepts	12
5. Notation Used to Specify Interface Functions	15
5.1 State Diagram Notation	15
5.2 Message Output Notation	17
6. Source Handshake Interface (SH) Function.....	17
6.1 General Description	17
6.2 Source Handshake Function State Diagram	18
6.3 Source Handshake Function State Descriptions	18
6.4 Source Handshake Function Allowable Subsets	21
6.5 Additional SH Function Requirements and Guidelines	22

C 1901-1987

7.	Acceptor Handshake Interface (AH) Function	22
7.1	General Description	22
7.2	Acceptor Handshake Function State Diagram	22
7.3	Acceptor Handshake Function State Descriptions	22
7.4	Acceptor Function Allowable Subsets	25
7.5	Additional AH Function Requirements and Guidelines	26
8.	Talker Interface (T) Function	26
8.1	General Description	26
8.2	Talker Function State Diagrams	27
8.3	Talker Function State Descriptions	30
8.4	Talker Function and Extended Talker Function Allowable Subsets	33
8.5	Additional T and TE Interface Function Requirements and Guidelines	35
9.	Listener Interface (L) Function	35
9.1	General Description	35
9.2	Listener Function State Diagram	35
9.3	Listener Function State Descriptions	37
9.4	Listener Function and Extended Listener Function Allowable Subsets	40
9.5	Additional L or LE Requirements and Guidelines	41
10.	Service Request Interface (SR) Function	41
10.1	General Description	41
10.2	Service Request Interface Function State Diagram	41
10.3	Service Request State Description	42
10.4	Service Request Interface Function Allowable Subsets	43
10.5	Additional SR Interface Function Requirements and Guidelines	43
11.	Remote/Local Interface (RL) Function	43
11.1	General Description	43

11.2	Remote/Local Function State Diagram	44
11.3	Remote/Local State Descriptions	45
11.4	Remote/Local Function Allowable Subsets	46
11.5	Additional RL Function Requirements and Guidelines	47
12.	Parallel Poll Interface (PP) Function	47
12.1	General Description	47
12.2	Parallel Poll Function State Diagram	48
12.3	Parallel Poll State Descriptions	49
12.4	Parallel Poll Interface Function Allowable Subsets	51
12.5	Additional PP Interface Function Requirements and Guidelines	52
13.	Device Clear Interface (DC) Function	52
13.1	General Description	52
13.2	Device Clear Function State Diagram	52
13.3	Device Clear Function State Descriptions	52
13.4	Device Clear Interface function Allowable Subsets	54
13.5	Additional DC Function Requirements and Guidelines	54
14.	Device Trigger Interface (DT) Function	54
14.1	General Description	54
14.2	Device Trigger Function State Diagram	54
14.3	Device Trigger Function State Descriptions	54
14.4	Device Trigger Interface Allowable Subsets	55
14.5	Additional DT Function Requirements and Guidelines	56
15.	Controller Interface (C) Function	56
15.1	General Description	56
15.2	Controller Function State Diagram	56
15.3	Controller State Descriptions	59
15.4	Controller Interface Function Allowable Subsets	66
15.5	Additional C Function Requirements and Guidelines	66

C 1901-1987

16. Remote Message Coding and Transfer	68
16.1 Remote Message Coding	68
16.2 Remote Message Coding Concepts	68
16.3 Remote Message Transfer	68
16.4 Remote Message Coding Table Organization and Conventions	69
16.5 Remote Message Coding Table Perspective	69
16.6 JIS Code Representation: Message Coding Guidelines	72
16.7 State Transition Timing Values	73
Section 3. Electrical Specifications	74
17. Application	74
18. Logical and Electrical State Relationships	74
19. Driver Requirements	74
19.1 Driver Types	75
19.2 Driver Specifications	75
20. Receiver Requirements	75
20.1 Receiver Specifications (Allowed)	75
20.2 Receiver Specifications (Preferred)	75
21. Composite Device Load Requirements	76
21.1 Resistive Termination	76
21.2 Negative Voltage Clamping	76
21.3 D.C. Load Requirements	76
21.4 Capacitive Load Limit	77
21.5 Typical Circuit Configuration	77
22. Ground Requirements	78
23. Cable Characteristics	79
23.1 Conductor Requirements	79
23.2 Cable Construction	79
24. State Transition Time Values	80

Section 4. Mechanical Specifications	82
25. Application	82
26. Connector Type	82
26.1 Electrical Requirements	82
26.2 Mechanical Requirements	82
26.3 Environmental Requirements	83
27. Contact Assignments	83
28. Device Connector Mounting	83
29. Cable Assembly	85
Section 5. System Applications and Guidelines for the Designer	87
30. System Compatibility	87
30.1 General Guidelines	87
31. Data Rate Consideration	87
32. Device Capabilities	88
32.1 Busy Function	88
32.2 Remote Local Applications	89
33. AND and OR Logic Operations	90
33.1 RFD and DAC Messages	90
33.2 SRQ Message	90
33.3 Circuit Implementations	91
34. Address Assignment	93
35. Typical Combinations of Interface Functions	93
Section 6. System Requirements and Guidelines for the User	94
36. System Compatibility	94
36.1 General Guidelines	94
37. System Installation Requirements	94
37.1 System Configuration Requirements	94

C 1901-1987

38. Address Assignment	95
38.1 Talk Addresses	95
38.2 Listen Addresses	95
38.3 Secondary Addresses	95
39. Cabling Restrictions	96
39.1 Maximum Cable Length	96
39.2 Distribution of Maximum Cable Lengths	96
39.3 Cabling Configurations	96
39.4 Ground Requirement Precaution	96
40. Operational Sequences	96
40.1 Data Transfer	97
40.2 Serial Poll	97
40.3 Control Passing	98
40.4 Parallel Poll	98
40.5 Placing Devices in Forced Remote Control	99
40.6 Sending Interface Clear	99
40.7 Restriction on the Use of tca	99
Appendix A. Typical Instrument System	100
Appendix B. Handshake Process Timing Sequence	102
Appendix C. Interface Function Allowable Subset Reference List and Function Identification Code	107
Appendix D. Interface Message Reference List	113
Appendix E. Multiline Interface Messages	116
Appendix F. Logic Circuit Implementation	117
Appendix G. Parallel Poll Sequence	119
Appendix H. Description of Interface Parameter on Data Sheet	120
Appendix J. Recommended Procedure to Ameliorate Shield Effect of the Cables Specified in This Standard	126
Reference. Index of Terms and Acronyms	128

JAPANESE INDUSTRIAL STANDARD

J I S

An Interface System for Programmable
Measuring Instruments

C 1901-1987

Section 1. General

1. Scope and Object1.1 Scope

1.1.1 This Standard is applicable to interface systems used to interconnect both programmable and non-programmable electronic measuring apparatus with other apparatus and accessories necessary to assemble instrumentation systems.

1.1.2 This Standard is applicable to the interface of instrumentation systems, or portions of them, in which:

- (1) the data exchanged among the interconnected apparatus is digital (as distinct from analogue);
- (2) the total transmission path length over the interconnecting cables does not exceed 20 m;
- (3) the data rate across the interface on any signal line does not exceed 1 megabit per second.

The basic functional specifications of this standard may be used in digital interface applications which require:

- (1) longer distances,
- (2) more devices,
- (3) increased noise immunity,
- (4) or combinations of these.

Different electrical and mechanical specifications may be required (e.g., symmetrical circuit configurations, high threshold logic, special connectors or cable configurations) for these extended applications.

1.1.3 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.

1.1.4 This Standard applies generally to laboratory and production test environments which are both electrically quiet and restricted as to physical dimensions (distances between the system components).

Applicable Standards, Corresponding International Standard and Reference Standards: See page 99.

1.1.5 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 standard may be also applied to interconnect the internal subsections within a self-contained equipment.

1.2 Object This standard is intended:

- (1) to define a general-purpose system for use in limited distance applications,
- (2) to specify the device-independent mechanical, electrical, and functional interface requirements which apparatus shall meet in order to be interconnected and communicate unambiguously via the system,
- (3) to specify the terminology and definitions related to the system,
- (4) to enable the interconnection of independently manufactured apparatus into a single functional system,
- (5) to permit apparatus with a wide range of capability - from the simple to the complex - to be interconnected to the system simultaneously,
- (6) to permit direct communication between apparatus without requiring all messages to be routed to a control or intermediate unit,
- (7) to define a system with a minimum of restrictions on the performance characteristics of apparatus connected to the system,
- (8) to define a system that permits asynchronous communication over a wide range of data rates,
- (9) to define a system which, of itself, may be relatively low-cost and permits the interconnection of low-cost devices,
- (10) to define a system that is easy to use.

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

Remark: For the latter two items, reference is made to IEC Publication 348: Safety Requirements for Electronic Measuring Apparatus, and IEC Publication 359: Expression of the Functional Performance of Electronic Measuring Equipment.

2. Definitions

Throughout this standard, and in so far as further distinction is not necessary:

- (1) the term "system" denotes the byte-serial, bit-parallel interface system that, in general, includes all circuits, cables, connectors, message repertoire, and control protocol to effect unambiguous data transfer between devices;

- (2) 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.

2.1 General System Terms

2.1.1 System A set of interconnected elements constituted to achieve a given objective by performing a specified function.

Remark: The system is considered to be separated from the environment and other external systems by an imaginary surface which cuts the links between them and the considered system. Through these links, the system is affected by the environment, is acted upon by external systems, or acts itself on the environment or the external systems.

2.1.2 Interface A common boundary between a considered system and another system, or between parts of a system, through which information is conveyed.

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

2.1.4 Programmable That characteristic of a device that makes it capable of accepting data to alter the state of its internal circuits to perform a specific task(s).

2.1.5 Remote Control A method whereby a device is programmable via its electrical interface connection in order to enable the device to perform different tasks.

2.1.6 Local Control A method whereby a device is programmable by means of its local (front or rear panel) control in order to enable the device to perform various tasks. (Also referred to as manual control.)

2.1.7 Compatibility The degree to which devices may be interconnected and used, without modification, when designed as defined throughout this standard (e.g. mechanical, electrical, functional).

2.1.8 Handshake Cycle The process whereby digital signals effect the transfer of each data character across the interface by means of an interlocked sequence of status and control signals. Interlocked denotes a fixed sequence of events in which one event in the sequence must occur before the next event may occur.

2.1.9 Programmable Measuring Apparatus A measuring apparatus that performs specified operations on command from the system and may transmit the results of the measurement(s) to the system.

2.1.10 Terminal Unit An apparatus that terminates the considered interface system and by means of which a connection (including code translation, if required) is made between the considered interface system and another external interface system.

2.2 Signals and Paths

2.2.1 Signal The physical representation of information.

Remark: For the purpose of this standard, this is a restricted definition of what is often called "signal" in the general sense, and hereinafter refers to digital electrical signals only.

2.2.2 Signal Parameter That parameter of an electrical quantity whose value or sequence of values conveys information.

2.2.3 Signal Level The magnitude of a signal compared to an arbitrary reference magnitude (voltage in the case of this standard).

2.2.3.1 High State The relatively more positive signal level used to assert a specific message content associated with one of two binary logic states.

2.2.3.2 Low State The relatively less positive signal level used to assert a specific message content associated with one of two binary logic states.

2.2.4 Signal Line One of a set of signal conductors in an interface system used to transfer messages among interconnected devices.

2.2.5 Byte A group of concurrent binary digits operated on as a unit and usually shorter than a computer word. (Frequently connotes a group of eight bits.)

2.2.6 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.

2.2.6.1 Unidirectional Bus A bus used by any individual device for one-way transmission of messages only, i.e. either input only or output only.

2.2.6.2 Bidirectional Bus A bus used by any individual device for two-way transmission of messages, i.e. both input and output.

2.2.7 Byte Serial A sequence of bit parallel data bytes used to carry information over a common bus.

2.2.8 Bit Parallel A set of concurrent data bits present on a like number of signal lines used to carry information. Bit parallel data bits may be acted upon concurrently as a group (byte) or independently as individual data bits.

3. Interface System Overview

3.1 Interface System Objective

3.1.1 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.

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

- (1) messages used to manage the interface system itself, hereinafter called interface messages;
- (2) 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.

Remark: The detailed specification of device-dependent messages is beyond the scope of this standard.

3.2 Fundamental Communication Capabilities

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

- (1) a device acting as a listener;
- (2) a device acting as a talker;
- (3) a device acting as a controller.

Remark: The terms talker and listener have specialized meanings in this standard to better describe the role of devices. These terms are used in order to avoid confusion with other technical terms such as transmitter and receiver.

3.2.2 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.

Remark: 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 Section 2 to distinguish between different types of controller capabilities used in the interface system.

3.2.3 Listener, talker, and controller capabilities occur individually or in any combination in devices interconnected via the interface system (see Fig. 1).

3.2.4 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 requires 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, on command from the controller. This may occur simultaneously with either selected devices or all devices in a system.
- (d) The remote/local function provides a device with the ability to select between programme data from the bus and local data (e.g. front panel controls).

3.3 Message Paths and Bus Structure

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

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

3.3.3 The bus structure is organized into the following three sets of signal lines. Fig. 1 illustrates the basic communication paths.

- (1) data bus, eight signal lines;
- (2) data byte transfer control bus, three signal lines;
- (3) interface management bus, five signal lines.

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

- (a) data input output 1 (DIO 1);
- ⋮
- (h) data input output 8 (DIO 8).

Message bytes are carried on the DIO signal lines:

- (1) in a bit-parallel byte-serial form;
- (2) asynchronously;
- (3) bidirectionally.

Remark: A message may be carried on an individual DIO signal line when required.

3.3.3.2 Data byte transfer control bus 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) Date 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.
- (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* process to transfer each data byte across the interface.

Note: Concept of three-wire handshake is described in Clauses 6. and 7.

Remark: As the industrial proprietorial right is endorsed on the three-wire handshake, see Item 2 in annexed explanation.

3.3.3.3 Interface management bus is 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 inter-connected 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 select between two alternate sources of device programming data.
- (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 parallel poll.

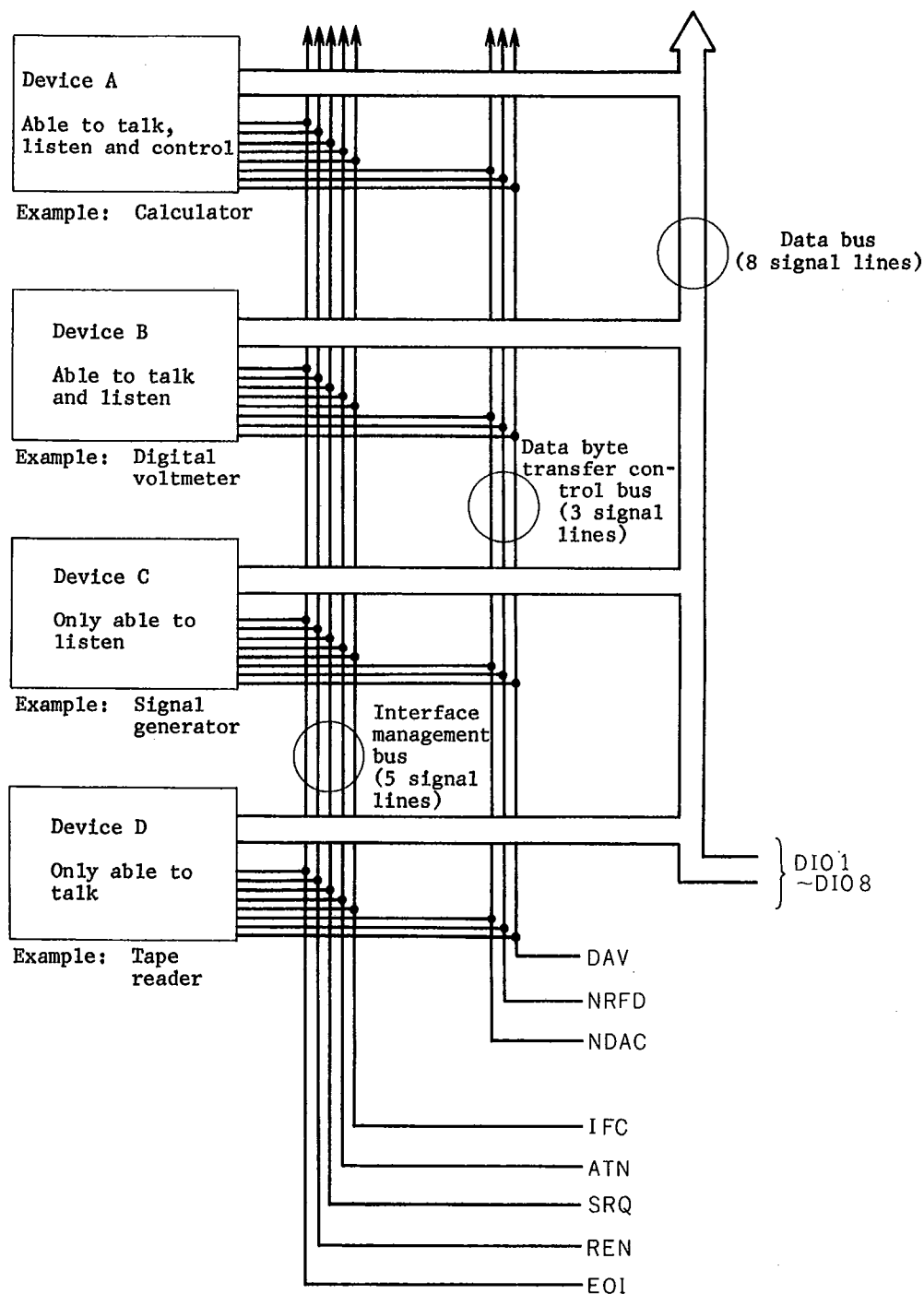
3.4 Interface System Elements

The primary elements of this interface system are:

- (1) functional elements;
- (2) electrical elements;
- (3) mechanical elements.

Each of the elements is described in the following section.

Fig. 1. Interface Capabilities and Bus Structure



Section 2. Functional Specifications

4. 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:

- (1) device functions (definition is application-dependent);
- (2) interface functions (definition is application-independent);
- (3) message coding logic.

All communication to or from interface functions is defined in terms of messages and state linkages (see Sub-clause 4.3).

All messages carried on the signal lines are coded according to the coding logic defined in Clause 16.

4.1 Device Functions The scope, purpose, size, content, and organization of the device function area (e.g. analogue signal measurement capability, range, modes of operation, etc.) are beyond the scope of this standard. Fig. 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.

4.2 Interface Function Concepts

4.2.1 Interface Function 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 each particular class of messages.

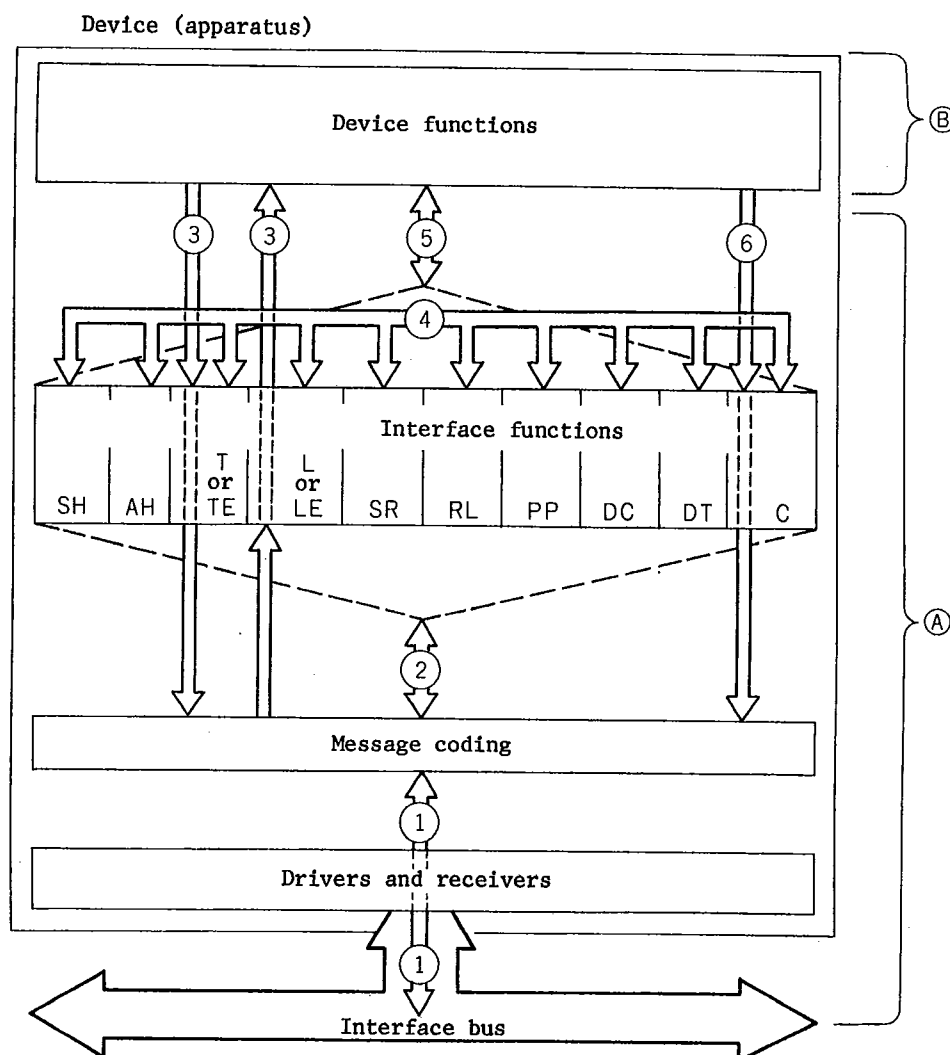
4.2.2 Interface Function State

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

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

4.2.2.3 For each state of an interface function definitions are given for: messages which may or must be sent over the interface while that state is active, and 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.

Fig. 2. Partition of Functions within a Device



where

- Ⓐ : capability defined by this standard
- Ⓑ : 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 in this standard; messages from interface functions are specified by the designer)
- ⑥ : remote interface messages sent by device functions within a controller

4.2.3 Interface Function Repertoire

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

Table 1

Interface function repertoire		
Interface function	Symbol	Relevant message paths
Source handshake	SH	1, 2, 4, 5
Acceptor handshake	AH	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

4.2.3.2 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.

4.2.4 Interface Function Assumptions and Perspective

4.2.4.1 The state diagrams used to define the interface functions do not indicate, either explicitly or implicitly, the intended existence of specific circuit 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 element.

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

4.2.4.3 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.

4.2.4.4 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. Sections 5 and 6 will describe the interaction among devices from the system perspective.

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

4.2.4.6 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.

4.3 Message Concepts

4.3.1 Message Each message represents a quantity of information and will be received either true or false at any specific time.

All communication between an interface function and its environment is accomplished through messages sent or received.

4.3.2 Local Message Path and Content

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

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

Remark: Certain local messages are conveyed as remote messages and vice versa.

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

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

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

4.3.3 Remote Message Route and Content

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

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

4.3.3.3 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 Fig. 2 message route ②.

4.3.3.4 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 Fig. 2, message route ③.

4.3.4 State Linkage Route and Content

4.3.4.1 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 Fig. 2, message route ④.

4.3.5 Message Coding

4.3.5.1 Message coding is the act of translating remote messages to or from interface signal line values.

4.3.5.2 A message sent over a single signal line is called a uniline message. Two or more of these messages can be sent concurrently.

4.3.5.3 A message which 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.

4.3.6 Classification of Multiline Messages

4.3.6.1 Multiline messages are interpreted as interface messages when the ATN message is true.

4.3.6.2 Multiline messages are interpreted as device-dependent messages when the ATN message is false.

4.3.6.3 Each device, when the ATN message is true, may accept and process any or all of these specific classes of multiline messages as determined both by the extent of the interface function capability contained within the device and the current set of active interface function states:

Interface messages are classified as follows:

- (1) universal commands (all devices independent of interface function states);
- (2) addressed commands (all devices addressed to listen);
- (3) addresses (all devices);
- (4) secondary addresses or commands (all devices enabled by a primary address or command).

For a list of specific commands, see Table 36.

4.3.6.4 Each device, when the ATN message is false, may accept and process its own device-dependent message when addressed to listen.

4.3.7 Message Transfer Conventions

4.3.7.1 Remote Message Transfer Conventions

- (a) The value (true or false) of all remote 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 the remote message coding Table 38.
- (c) Since normal interface operation allows two or more devices to simultaneously send opposite values of the remote messages, a technique must 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 passive. Messages must 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 (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 (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 Clause 33).
- (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 line (see Clause 33).
- (h) Only the multiline message(s) to be sent true will be specified for an interface function state since 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 that state is active.

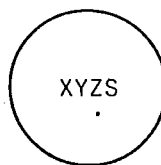
4.3.7.2 Local Message Transfer Conventions

- (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.

5. Notation Used to Specify Interface Functions

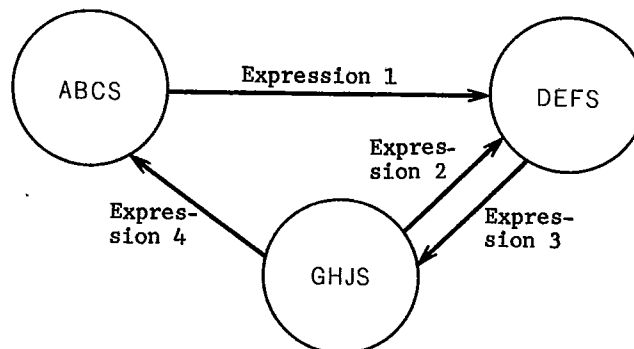
5.1 State Diagram Notation

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



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

5.1.3 Each transition is qualified by an expression whose value is either true or false. The interface function must remain in its current state if all expressions which qualify transitions leading to other states are false. The interface function must 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.



5.1.3.1 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.

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

5.1.3.3 A remote message (received via the interface) is represented by a three-letter mnemonic written in upper case: for example ATN.

5.1.3.4 A linkage from another state diagram is represented by a four-letter mnemonic enclosed in an oval: for example LACS. A state linkage is true if the enclosed state is currently active; otherwise, it is false.

5.1.3.5 A minimum time-limit is represented by the symbol "Tn". This symbol achieves a true value only after the interface has been in the state originating the corresponding transition for the time value specified. The values for these time-limits are contained in Table 39.

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

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

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

5.1.3.9 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.

5.1.4 If a transition is further qualified by a maximum time-limit "(within Tn)", then the state pointed to must be active within the specified amount of time after the expression becomes true.

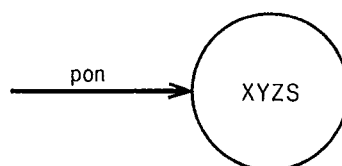
5.1.5 If a portion of an expression is optional in that its true value is not required for the complete expression to be true (at the designer's choice), then it is enclosed within square brackets "[...]".

5.1.6 If a specific expression causes a transition to a state from all other states of the diagram, a 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 is assumed to originate in all states (i.e. $\overline{\text{IFC}}$ or $\overline{\text{REN}}$). It is further assumed that these same expressions are false (i.e. IFC or REN) to permit all other transitions in the diagram, and therefore they are omitted from the diagrams.

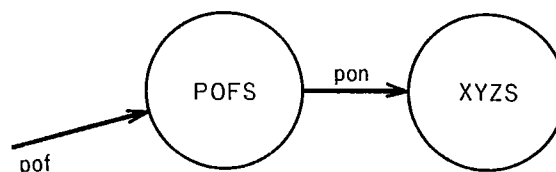
expression $\rightarrow$ XYZS

5.1.7 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 active at "power-on" time, a shorthand form is used show in the "pon".

(1) abbreviated notation used on state diagram:



- (2) complete representation implied by above symbol:



5.2 Message Output Notation

5.2.1 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.

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

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

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

- (1) "T" indicates active true.
- (2) "F" indicates active false.
- (3) "(T)" indicates passive true.
- (4) "(F)" indicates passive false.

5.2.5 One column in each table is allocated, if require, 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 since multiline messages are mutually exclusive. Parentheses around a multiline message name specify that it must be sent passive rather than active true.

5.2.6 A separate column for device function interaction summarizes the corresponding types of messages device functions are allowed to send or receive. Local messages 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.

6. Source Handshake Interface (SH) Function

6.1 General Description The SH function provides a device with the capability to guarantee the proper transfer of multiline messages. An interlocked handshake sequence between the SH function and one or more Acceptor Handshake Interface Functions (each contained within separate devices) guarantees asynchronous transfer of each multiline message. The SH function controls the initiation of, and termination of, the transfer of a multiline message byte. This function utilizes the DAV, RFD, and DAC messages to effect each message byte transfer.

6.2 Source Handshake Function State Diagram The SH function shall be implemented so as to perform according to the state diagram given in Fig. 3, and the state descriptions given throughout Clause 6. Table 2 specifies the set of messages and states required to effect transition from one active state to another. Table 3 specifies the messages that shall be sent and the device function interaction required while each state is active.

6.3 Source Handshake Function State Descriptions

6.3.1 Source Idle State (SIDS)

6.3.1.1 In the SIDS state the SH function is not engaged in the handshake cycle and does not have a new message byte available.

6.3.1.2 In the SIDS state the DAV message shall be sent passive false.

6.3.1.3 The SH function shall exit the SIDS state and enter the Source Generate State (SIGN) if:

- (1) the Talker Active State (TACS) is active;
- (2) or the Serial Poll Active State (SPAS) is active;
- (3) or the Controller Active State (CACS) is active.

Fig. 3. Source Handshake State Diagram

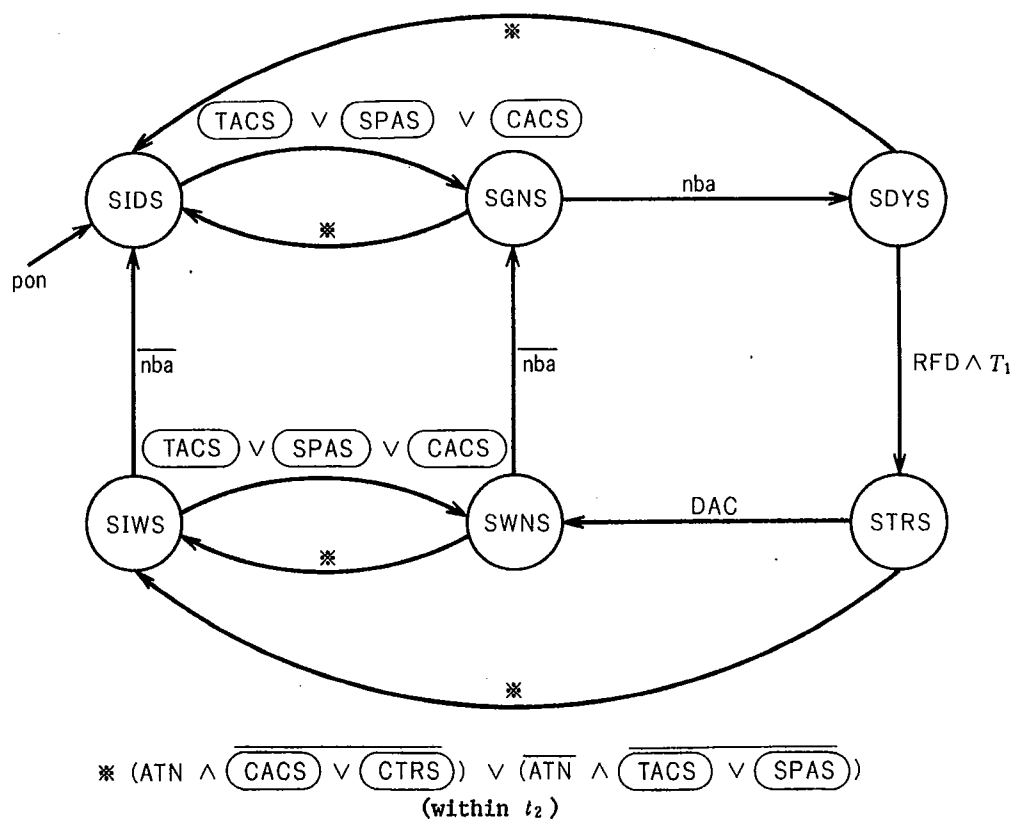


Table 2. SH Mnemonics

Messages:	Interface States:
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. SH Message Outputs

SH state	Remote message sent	Device function interaction
	DAV	
SIDS	(F)	Device function can change remote multiline messages
SGNS	F	Device function 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	Device function requested to change multiline messages
SIWS	(F)	Device function requested to change multiline messages

6.3.2 Source Generate State (SGNS)

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

6.3.2.2 In the SGNS state the SH function shall send the Data Valid (DAV) message false. In this state the device may change the multiline message being sent via the Talker or Controller Interface Function while in the TACS or CACS states.

6.3.2.3 The SH function shall exit the SGNS state and enter:

- (a) the Source Delay State (SDYS) if the new byte available (nba) message is true:

(b) the SIDS state within t_2 if either:

- (1) the Attention (ATN) message is true and neither the CACS state nor the CTRS state is active;
- (2) or the ATN message is false and neither the TACS state nor the SPAS state is active.

6.3.3 Source Delay State (SDYS)

6.3.3.1 In the SDYS state the SH function is waiting for a message byte to settle on the interface signal lines after the change during the SGNS state and for all the Acceptor Functions to indicate their readiness to accept the message byte.

6.3.3.2 In the SDYS state the SH function shall send the DAV message false. In this state the device shall not change the multiline message being sent.

6.3.3.3 The SH function shall exit the SDYS state and enter:

- (a) the Source Transfer State (STRS) only after T_1 if the Ready For Data (RFD) message is true;
- (b) the SIDS state within t_2 if either:
 - (1) the Attention (ATN) message is true and neither the CACS state nor the CTRS state is active;
 - (2) or the ATN message is false and neither the TACS state nor the SPAS state is active.

6.3.4 Source Transfer State (STRS)

6.3.4.1 In the STRS state the SH function indicates to the acceptor handshake function that it is continuously sending a valid message byte.

6.3.4.2 In the STRS state the SH function shall send the DAV message true. In this state the device shall not change either the multiline message or the END message (if used) being sent.

6.3.4.3 The SH function shall exit the STRS state and enter:

- (a) the Source Idle Wait State (SIWS) with t_2 if either:
 - (1) the Attention (ATN) message is true and neither the CACS state nor the CTRS state is active;
 - (2) or the ATN message is false and neither the TACS state nor the SPAS state is active;

Remark: This implies an asynchronous interrupt (see Sub-clause 15.5).

- (b) the Source Wait For New Cycle State (SWNS) if the Data Accepted (DAC) message is true.

6.3.5 Source Wait For New Cycle State (SWNS)

6.3.5.1 In the SWNS state the SH function is waiting for the device to start a new message generation cycle.

6.3.5.2 In the SWNS state the SH function may send the DAV message true or false. In this state the device may change the multiline message being sent.

6.3.5.3 The SH function shall exit the SWNS state and enter:

- (a) the SGNS state if the nba message is false;
- (b) the SIWS state within t_2 if either:
 - (1) the Attention (ATN) message is true and neither the CACS state not the CTRS state is active;
 - (2) or the ATN message is false and neither the TACS state not the SPAS state is active.

6.3.6 Source Idle Wait State (SIWS)

6.3.6.1 In the SIWS state the SH 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 state allows a sequence of message byte transfers to be interrupted without loss of data over the interface while at the same time the device may continue to prepare for the new (next) message byte generation cycle.

6.3.6.2 In the SIWS state the DAV message shall be sent passive false.

6.3.6.3 The SH function shall exit the SIWS state and enter:

- (a) the SIDS state if the nba message is false;
- (b) the SWNS state if either the TACS state is active; or the SPAS state is active; or the CACS state is active.

6.4 Source Handshake Function Allowable Subsets The only allowable subsets to the SH function shall be those listed in Table 4.

Table 4

Identifi- cation	Description	States omitted	Other require- ments	Other function subsets required
SH $\emptyset$	No capability	All	None	None
SH 1	Complete capability	None	None	T 1 - T 8 or TE 1 - TE 8 or C 5 - C 28

6.5 Additional SH Function Requirements and Guidelines

6.5.1 Interpretation of the Local nba Messages

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

6.5.1.2 The nba message shall become true only in the SIDS or SGNS states. The nba message may become false in any other SH states.

6.5.2 Alternate Transition to Idle States The expression $(\overline{\text{ATN}} \wedge (\overline{\text{CACS}} \vee \overline{\text{CTRS}})) \vee (\overline{\text{ATN}} \wedge (\overline{\text{TACS}} \vee \overline{\text{SPAS}}))$ for interruption in Fig. 3 may be substituted by $\overline{\text{TACS}} \wedge \overline{\text{SPAS}} \wedge \overline{\text{CACS}} \wedge \overline{\text{CTRS}}$ if the transition of the latter expression can be effected within t_2 after the change of ATN.

7. Acceptor Handshake Interface (AH) Function

7.1 General Description The AH function provides a device with the capability to guarantee proper reception of remote multiline messages. An interlocked handshake sequence between an SH function and one or more AH functions (each contained within separate devices) guarantees asynchronous transfer of each message byte. An 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.

7.2 Acceptor Handshake Function State Diagram The AH function shall be implemented so as to perform according to the state diagram given in Fig. 4, and the state descriptions given throughout Clause 7. Table 5 specifies the set of messages and states required to effect transition from one active state to another. Table 6 specifies the messages that shall be sent and the device function interaction required while each state is active.

7.3. Acceptor Handshake Function State Descriptions

7.3.1 Acceptor Idle State (AIDS)

7.3.1.1 In the AIDS state the AH function is inactive and not engaged in the handshake cycle. The AH function powers on in the AIDS state.

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

7.3.1.3 The AH function shall exit the AIDS state and enter the Acceptor Not Ready State (ANRS) within t_2 if either:

- (1) the ATN message is true;
- (2) or the LACS state is active;
- (3) or the LADS state is active.

7.3.2 Acceptor Not Ready State (ANRS)

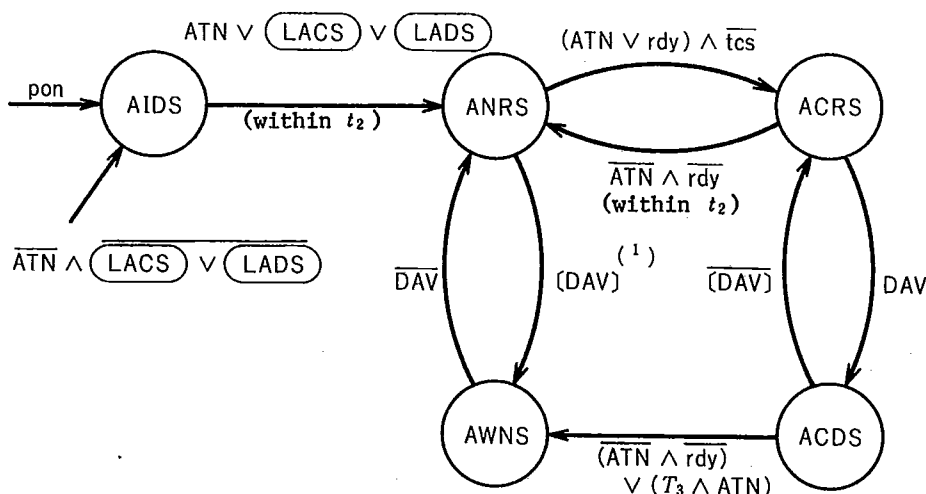
7.3.2.1 In the ANRS state the AH function indicates to the interface it has not yet prepared internally to continue with the handshake cycle.

7.3.2.2 In th ANRS state the RFD and DAC messages shall be sent false.

7.3.2.3 The AH function shall exit the ANRS state and enter:

- (a) the ACRS state if the take control synchronously (tcs) message is false (see Sub-clause 15.3.7.1) and either the ATN message is true or the rdy message is true;
- (b) the AIDS state if the ATN message is false and neither the LADS state is active nor the LACS state is active;
- (c) the AWNS state if the DAV message is true (optional, note that this transition will never occur under normal interface operation).

Fig. 4. Acceptor Handshake State Diagram



Note ⁽¹⁾ This transition will never occur under normal interface operation; however it may be implemented to simplify the interface function design.

Table 5. AH Mnemonics

Messages:	Interface States:
pon - power on	AIDS - ACCEPTOR IDLE STATE
rdy - ready for next message	ANRS - ACCEPTOR NOT READY STATE
tes - take control synchronously ⁽¹⁾	ACRS - ACCEPTOR READY STATE
	ACDS - ACCEPT DATA STATE
ATN - ATTENTION	AWNS - ACCEPTOR WAIT FOR NEW CYCLE STATE
DAV - DATA VALID	
	(LADS) - LISTENER ADDRESSED STATE (L function)
	(LACS) - LISTENER ACTIVE STATE (L function)

Note ⁽¹⁾ See Sub-clause 15.3.7.1.

Table 7. AH Message Outputs

AH state	Remote message sent		Device function interaction
	RFD	DAC	
AIDS	(T)	(T)	Device function can receive neither remote multiline nor END messages
ANRS	F	F	Device function can receive neither remote multiline nor END messages
ACRS	(T)	F	Device function can receive neither remote multiline nor END messages
AWNS	F	(T)	Device function can receive neither remote multiline nor END messages
ACDS	F	F	Device function receive remote multiline messages if LACS is active

7.3.3 Acceptor Ready State (ACRS)

7.3.3.1 In the ACRS state the AH function indicates to the interface that it is prepared to receive multiline messages.

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

7.3.3.3 The AH function shall exit the ACRS state and enter:

- (a) the Accept Data State (ACDS) if the DAV message is true;
- (b) the AIDS state if the ATN message is false and neither the LADS state is active nor the LACS state is active;
- (c) the ANRS state within t_2 if both the ATN and the ready for next message (rdy) messages are false.

7.3.4 Accept Data State (ACDS)

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

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

7.3.4.3 The AH function shall exit the ACDS state and enter:

- (a) the Acceptor Wait For New Cycle State (AWNS) if either the ATN message is true and a period of T_3 has elapsed; or the AN and rdy messages are both false;
- (b) the AIDS state if the ATN message is false and neither the LADS state is active nor the LACS state is active;
- (c) the ACRS state if the DAV message is false.

7.3.5 Acceptor Wait For New Cycle State (AWNS)

7.3.5.1 In the AWNS state the AH function indicates that it has received a multiline message byte.

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

7.3.5.3 The AH function shall exit the AWNS state and enter:

- (a) the Acceptor Not Ready State (ANRS) if DAV is false;
- (b) the AIDS state if the ATN message is false and neither the LADS state is active nor the LACS state is active.

7.4 Acceptor Function Allowable Subsets The only allowable subsets to the AH function shall be those listed in Table 7.

Table 7

Identifi- cation	Description	States omitted	Other requirements	Other function subsets required
AH 0	No capability	All	None	None
AH 1	Complete capability	None	None	None

7.5 Additional AH Function Requirements and Guidelines

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

7.5.2 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 Clause 33.

7.5.3 Since 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 which exist in an implementation.

One consequence of this statement is that interface function state transitions which 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 must occur while the interface message is being received (transition abiding by the state diagrams).

If this type of implementation is chosen, then the AH functions should remain in the ANRS state even though the exit condition is true in order to hold the RFD message false (this is allowed by Sub-clause 4.2.4.6).

8. Talker Interface (T) Function (includes serial poll capabilities)

8.1 General Description

8.1.1 The T 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. The capability exists only when the T function is addressed to talk.

8.1.2 There are two alternative versions of the functions: one with and one without address extension. The normal T function uses a one-byte address. The T function with address extension (hereinafter referred to as the "Extended Talker (TE) function") uses a two-byte address. In all other respects, the capabilities of both versions are the same.

8.1.3 Only one of the two alternative Talker functions need be implemented in a specific device.

Remark: Both the T function and TE function are described concurrently throughout Clause 8 due to the extensive similarity between these two functions.

8.2 Talker Function State Diagrams

8.2.1 The T function shall be implemented so as to perform according to the state diagrams given in Fig. 5 and the state descriptions given throughout Clause 8. Table 8 specifies the set of messages and states required to effect transition from one active state to another. Table 9 specifies the messages that shall be sent and the device function interaction required while each state is active.

8.2.2 The TE function shall be implemented so as to perform according to the state diagrams given in Fig. 6 and the state descriptions given throughout Clause 8. Table 10 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.

Fig. 5. Talker State Diagram

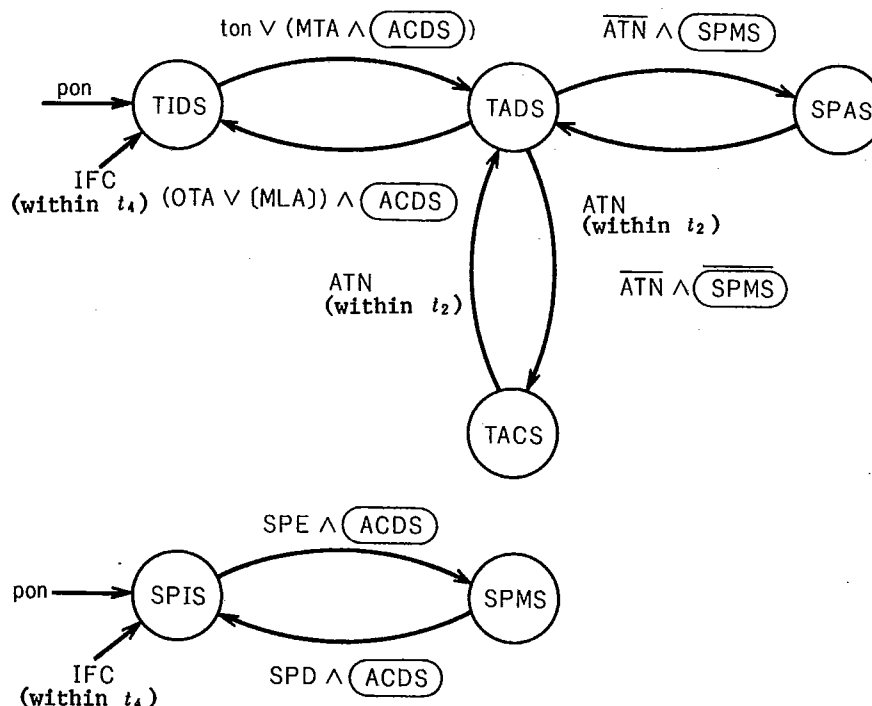


Table 8. T Mnemonics

Messages:	Interface States:
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 9. T or TE Message Outputs

T state	Qualifier	Remote message sent (2)			Device function interaction (DF)
		Multiline	END	RQS(4)	
TIDS		(NUL)	(F)	(F)	Device function not allowed to send messages
TADS		(NUL)	(F)	(F)	Device function not allowed to send messages
TACS		DAB(1) or EOS(1)	T(1) or F(1)	(F)	Device function can send DAB, EOS, or END message(s) concurrent with DAB
SPAS	APRS(1) inactive	STB	F or T	F	Device function can send one STB message(3).
SPAS	APRS inactive	STB(1)	F or T	T	Device function can send one STB message(3).

Notes (1) Messages enable by the T function originating within the device functions.

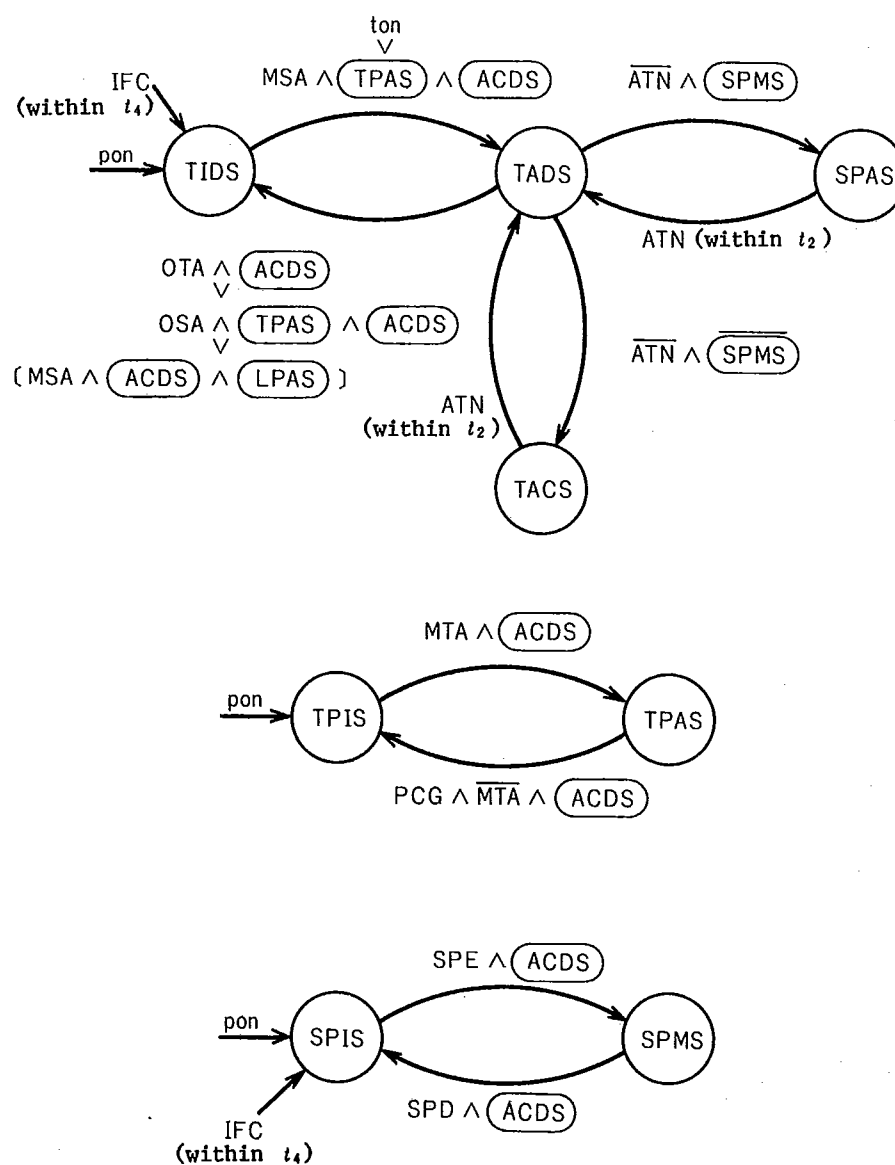
(2) See Table 38 (Clause 16).

(3) Under SH control.

(4) See Sub-clause 8.3.4.1.

(5) See Sub-clause 10.3.3.

Fig. 6. Extended Talker State Diagram



Remark: If the TE function is used together with the L function then $[MSA \wedge \overline{ACDS} \wedge \overline{LPAS}]$ shall be replaced by $[MLA \wedge \overline{ACDS}]$.

Table 10. TE Mnemonics

Messages:	Interface States:
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
MSA - MY SECONDARY ADDRESS	(L function)

8.3 Talker Function State Descriptions

8.3.1 Talker Idle State (TIDS)

8.3.1.1 In the TIDS state either the T function or the TE function is not engaged in sending data or status bytes.

The T function or the TE function powers on in the TIDS state.

8.3.1.2 In the TIDS state the END and RQS messages shall be sent passive false and the NUL message shall be sent passive true.

8.3.1.3 The T function shall exit the TIDS state and enter the Talker Addressed State (TADS) if either:

- (1) the My Talk Address (MTA) message is true and the ACDS state is active;
- (2) or the talk only (ton) message is true (see Sub-clause 8.5.3).

8.3.1.4 The TE function shall exit the TIDS state and enter the TADS state if either:

- (1) the My Secondary Address (MSA) message is true, and the ACDS state is active, and the Talker Primary Address State (TPAS) is active;
- (2) or the ton message is true.

8.3.2 Talker Addressed State (TADS)

8.3.2.1 In the TADS state the T function has received its talk address and is prepared for, but not engaged in, sending data or status bytes. In the TADS state 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.

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

8.3.2.3 The T function shall exit the TADS state 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 the SPMS state is active;
- (c) the TIDS state if either:
 - (1) the Other Talk Address (OTA) message is true and the ACDS state is active;
 - (2) or the MLA message is true and the ACDS state is active;
 - (3) or within t_4 if the IFC message is true.

Remark: Use of the MLA message is optional.

8.3.2.4 The TE function shall exit the TADS state and enter:

- (a) the TACS state if the ATN message is false and the SPMS state is inactive;
- (b) the SPAS state if the ATN message is false and the SPMS state is active;
- (c) the TIDS state if either:
 - (1) the OTA message is true and the CDS state is active;
 - (2) or the Other Secondary Address (OSA) message is true and the TPAS and ACDS states are active;
 - (3) or the MSA message is true and both the Listener Primary Address State (LPAS) and ACDS states are active;
 - (4) or within t_4 if the Interface Clear (IFC) message is true.

Remark: Use of the expression containing the MSA message is optional. See also Remark under Fig. 6.

8.3.3 Talker Active State (TACS)

8.3.3.1 In the TACS state the T function, or the TE function, enables the transfer of the device message (DAB, EOS, STB, END, etc.) from the device function to DIO 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 (including the determination of true or false of END if used).

8.3.3.2 During the TACS state the Data Byte (DAB) or End of String (EOS) and END messages may be sent by the device functions.

The RQS message shall be sent passive false.

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

8.3.3.3 The T function or the TE function shall exit the TACS state and enter:

- (a) the TADS state within t_2 if the ATN message is true;
- (b) the TIDS state within t_4 if the IFC message is true.

8.3.4 Serial Poll Active State (SPAS)

8.3.4.1 In the SPAS state the T function or the TE function enables the transfer of a single status message from the device function to the DIO lines using the SH function to control the transfer of the status byte that contains both the RQS and STB messages. Although a controller needs only one byte 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 each bit except the first one byte may change between subsequent transfers although the RQS message is held unaltered by the SR function.

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

Remark: The PRS state is contained in the Service Request Interface Function.

8.3.4.3 The T function or the TE function shall exit the SPAS state and enter:

- (a) the TADS state within t_2 if the ATN message is true;
- (b) the TIDS state within t_4 if the IFC message is true.

8.3.5 Serial Poll Idle State (SPIS)

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

8.3.5.2 The SPIS state does not provide a remote message sending capability.

8.3.5.3 The T function or the TE function shall exit the SPIS state and enter the SPMS state if the Serial Poll Enable (SPE) message is true and the ACDS state is active.

8.3.6 Serial Poll Mode State (SPMS)

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

8.3.6.2 The SPMS state does not provide a remote message-sending capability.

8.3.6.3 The T function or the TE function shall exit the SPMS state and enter the SPIS state if either:

- (1) the Serial Poll Disable (SPD) message is true and the ACDS state is active;
- (2) or within t_4 if the IFC message is true.

8.3.7 Talker Primary Idle State (TPIS)

8.3.7.1 In the TPIS state the TE function is able to recognize its primary address but not able to respond to its secondary address. The TE function powers on in the TPIS state.

8.3.7.2 The TPIS state does not provide a remote message-sending capability.

8.3.7.3 The TE function shall exit the TPIS state and enter the TPAS state if the MTA message is true and the ACDS state is active.

8.3.8 Talker Primary Addressed State (TPAS)

8.3.8.1 In the TPAS state the TE function is able to recognize and respond to its secondary address.

8.3.8.2 The TPAS state does not provide a remote message-sending capability.

8.3.8.3 The TE function shall exit the TPAS state and enter the TPIS state if the Primary Command Group (PCG) message is true, and the MTA message is false, and the ACDS state is active.

8.4 Talker Function and Extended Talker Function Allowable Subsets
The only allowable subsets to the talker and extended talker interface functions shall be those listed in Tables 11 and 12.

34
C 1901-1987

Table 11

Identification	Description					States omitted	Other requirements	Other function subsets required
	Capabilities							
	Basic talker	Serial poll	Take only mode	Unaddress if MLA				
T0	N	N	N	N		All	None	None
T1	Y	Y	Y	N		None	Omit (MLA $\wedge$ (ACDS))	SH 1 and AH 1
T2	Y	Y	N	N		None	1. Omit (MLA $\wedge$ (ACDS)) 2. ton always false	SH 1 and AH 1
T3	Y	N	Y	N		SPIS, SPMS, SPAS	Omit (MLA $\wedge$ (ACDS))	SH 1 and AH 1
T4	Y	N	N	N		SPIS, SPMS, SPAS	1. Omit (MLA $\wedge$ (ACDS)) 2. ton always false	SH 1 and AH 1
T5	Y	Y	Y	Y		None	Include (MLA $\wedge$ (ACDS))	SH 1 and (L 1 to L 4 or LE 1 to LE 4)
T6	Y	Y	N	Y		None	1. Include (MLA $\wedge$ (ACDS)) 2. ton always false	SH 1 and (L 1 to L 4 or LE 1 to LE 4)
T7	Y	N	Y	Y		SPIS, SPMS, SPAS	Include (MLA $\wedge$ (ACDS))	SH 1 and (L 1 to L 4 or LE 1 to LE 4)
T8	Y	N	N	Y		SPIS, SPMS, SPAS	1. Include (MLA $\wedge$ (ACDS)) 2. ton always false	SH 1 and (L 1 to L 4 or LE 1 to LE 4)

Table 12

Identification	Description					States omitted	Other requirements	Other function subsets required
	Capabilities							
	Basic extended talker	Serial poll	Talk only mode	Unaddress if MSA $\wedge$ (LPAS)				
TE 0	N	N	N	N		All	None	None
TE 1	Y	Y	Y	N		None	Omit (MSA $\wedge$ (LPAS) $\wedge$ (ACDS))	SH 1 and AH 1
TE 2	Y	Y	N	N		None	1. Omit (MSA $\wedge$ (LPAS) $\wedge$ (ACDS)) 2. ton always false	SH 1 and AH 1
TE 3	Y	N	Y	N		SPIS, SPMS, SPAS	Omit (MSA $\wedge$ (LPAS) $\wedge$ (ACDS))	SH 1 and AH 1
TE 4	Y	N	N	N		SPIS, SPMS, SPAS	1. Omit (MSA $\wedge$ (LPAS) $\wedge$ (ACDS)) 2. ton always false	SH 1 and AH 1
TE 5	Y	Y	Y	Y		None	Include (MSA $\wedge$ (LPAS) $\wedge$ (ACDS))	SH 1 and (L 1 to L 4 or LE 1 to LE 4)
TE 6	Y	Y	N	Y		None	1. Include (MSA $\wedge$ (LPAS) $\wedge$ (ACDS)) 2. ton always false	SH 1 and (L 1 to L 4 or LE 1 to LE 4)
TE 7	Y	N	Y	Y		SPIS, SPMS, SPAS	Include (MSA $\wedge$ (LPAS) $\wedge$ (ACDS))	SH 1 and (L 1 to L 4 or LE 1 to LE 4)
TE 8	Y	N	N	Y		SPIS, SPMS, SPAS	1. Include (MSA $\wedge$ (LPAS) $\wedge$ (ACDS)) 2. ton always false	SH 1 and (L 1 to L 4 or LE 1 to LE 4)

8.5 Additional T and TE Interface Function Requirements and Guidelines

8.5.1 Each device which includes a T function or a TE function shall provide a means by which the talk address (or secondary address) which it recognizes as MTA (or MSA) can be changed in the field by the user of the device.

8.5.2 The interruption of a device sending data by transition in and out of the TACS state should not adversely affect the format of the output data. It is recommended that a device returning to the TACS state should continue with the output data string at the point of interruption.

8.5.3 Each device which includes the ton message shall be provided with a manual switch to generate the Talk Only mode.

9. Listener Interface (L) Function

9.1 General Description

9.1.1 The L function provides a device with the capability to receive device-dependent data (including status data) over the interface from other devices. This capability exists only when the L function is addressed to listen.

9.1.2 There are two alternative versions of the function: one with and one without address extension. The normal L function uses a one-byte address. The L function with address extension, (hereinafter called as an "LE function") uses a two-byte address. In all other respects, the capabilities of both versions are the same.

9.1.3 Only one of the two alternative L functions need be implemented in a specified device.

Remark: Both the L function and LE function are described concurrently throughout Clause 9 due to the extensive similarity between these two functions.

9.2 Listener Function State Diagram

9.2.1 The L function shall be implemented so as to perform according to the state diagram given in Fig. 7, and the state descriptions given throughout Clause 9. Table 13 specifies the set of messages and states required to effect transition from one active state to another. Table 14 describes the device function interaction required while each state is active.

9.2.2 The LE function shall be implemented so as to perform according to the state diagram in Fig. 8, and the state descriptions given throughout Clause 9. Table 15 specifies the set of messages and states required to effect transition from one active state to another. Table 14 describes the device function interaction required while each state is active.

Fig. 7. Listener State Diagram

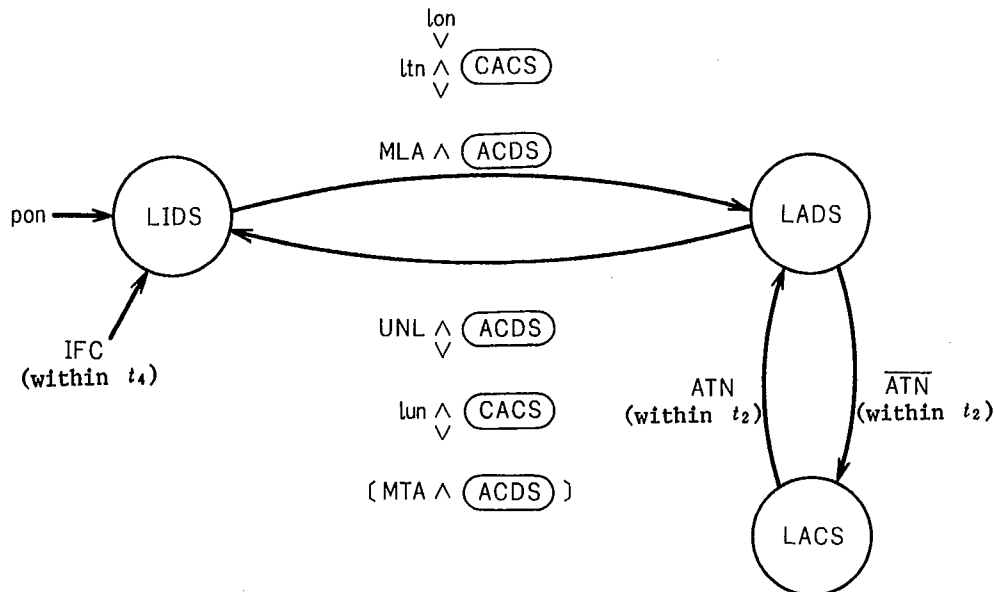


Table 13. L Mnemonics

Messages:	Interface States:
pon - power on	LIDS - LISTENER IDLE STATE
ltn - listen	LADS - LISTENER ADDRESSED STATE
lun - local unlisten	LACS - LISTENER ACTIVE STATE
lon - listen only	
IFC - INTERFACE CLEAR	ACDS - ACCEPT DATA STATE (AH function)
ATE - ATTENTION	CACS - CONTROLLER ACTIVE STATE (C function)
UNL - UNLISTEN	
MLA - MY LISTEN ADDRESS	
MTA - MY TALK ADDRESS	

Table 14. L or LE Message Outputs

L or LE state	Remote message sent	Device function interaction
LIDS	None	Device function not allowed to receive messages
LADS		Device function not allowed to receive messages
LACS		Device function can receive one device-dependent message byte each time ACDS state is active