IEEE Standard for a Smart Transducer Interface for Sensors and Actuators—Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats

Sponsor
TC-9, Committee on Sensor Technology
of the
IEEE Instrumentation and Measurement Society

Approved 16 September 1997

IEEE Standards Board

Abstract: A digital interface for connecting transducers to microprocessors is defined. A TEDS and its data formats are described. An electrical interface, read and write logic functions to access the TEDS and a wide variety of transducers are defined. This standard does not specify signal conditioning, signal conversion, or how the TEDS data is used in applications.
Keywords: communication protocol, digital interface, microprocessor, NCAP, sensor interface, smart actuator, smart sensor, smart sensor interface, smart transducer, smart transducer interface, STIM, TEDS, 1451

The Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street, New York, NY 10017-2394, USA

Copyright © 1998 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 1998. Printed in the United States of America.

ISBN 1-55937-963-4

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

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE that have expressed an interest in participating in the development of the standard.

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

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

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 all 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 on standards and requests for interpretations should be addressed to:

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

Note: Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

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

Introduction

[This introduction is not part of IEEE Std 1451.2-1997, IEEE Standard for a Smart Transducer Interface for Sensors and Actuators—Transducer to Microprocessor Communications Protocols and Transducer Electronic Data Sheet (TEDS) Formats.]

The main objectives of this standard are to:

— Enable plug and play at the transducer (sensor or actuator) level by providing a common communication interface for transducers.
— Enable and simplify the creation of networked smart transducers.
— Facilitate the support of multiple networks.

The existing fragmented sensor market is seeking ways to build low-cost, networked smart sensors. Many sensor network or fieldbus implementations are currently available, each with its own strengths and weaknesses for a specific application class. Interfacing transducers to all these control networks and supporting the wide variety of protocols represents a significant and costly effort to transducer manufacturers. A universally-accepted transducer interface standard would not only allow for the development of smart sensors and actuators, it could also lead to lower development costs. Therefore, the objective of this project is not to propose another control network, but to develop a smart transducer interface standard that will isolate the choice of transducers from the choice of networks. This would relieve the burden from the manufacturer of supporting a cross product of sensors versus networks, and would help to preserve the user's investment if it becomes necessary to migrate to a different network standard.

There is currently no defined common digital communication interface standard between transducers and network capable application processors (NCAPs). Each transducer manufacturer builds its own interface. Consequently, transducer manufacturers cannot afford to support all of the control networks for which their products may be suited. It was concluded at a series of five transducer interface workshops held between 1994 and 1995 that a common transducer communication interface standard be proposed. This common interface would allow the transducer manufacturers to more easily support multiple control networks.

This standard will simplify the development of networked transducers by defining hardware and software blocks that do not depend on specific control networks. This project has developed a standard hardware interface to connect a smart transducer interface module (STIM) to an NCAP. While the project does not include specifications for signal conditioning or data conversion, it does provide a mechanism for specifying the combination of transducer, signal conditioning, and signal conversion to the rest of the system. This mechanism is the transducer electronic data sheet (TEDS). The working group has defined a TEDS which supports a wide variety of transducers as well as a digital interface to access the TEDS, read sensors, and set actuators. This allows transducer manufacturers competitive differentiation in areas of quality, feature set and cost, and at the same time affords the opportunity to design to a common interface which can be used in a wide variety of applications.

The TEDS, which provides for self-identifying transducers, is at the core of this effort. The TEDS contains fields that fully describe the type, operation, and attributes of one or more transducers. By requiring that the TEDS be physically associated with the transducer, the resulting hardware partition encapsulates the measurement aspects in a STIM on one side of the digital interface, and the application related aspects on the NCAP on the other side. In addition to control networks, STIMs can be used with microprocessors in a variety of applications such as portable instruments and data acquisition cards.

Data output by the STIM may be in integer, single precision real, or double precision real formats. The data is passed to the NCAP and from the NCAP to the rest of the system. Further processing of this data may take place both in the NCAP and in other processors in the larger system. Throughout this standard it is assumed, but not required, that all processing will be performed on data in a single- or double-precision real format.

Copyright © 1998 IEEE. All rights reserved.

All fields in the TEDS are specified based on the assumption that, unless specifically stated to the contrary, all data will be converted to single- or double-precision real before any processing is performed.

This standard provides areas that are "open to industry." It should be noted that any use of these areas compromises the "plug and play" potential of NCAPs and STIMs.

The IEEE 1451.2 transducer to microprocessor interface is compatible with the P1451.1[1] information model standard. The two parts form a standard interface for networked smart sensors and actuators.

At the time this standard was approved, the Transducer to Microprocessor Communication working group had the following membership:

Stan P. Woods, *Chair* **Larry Malchodi,** *Vice Chair*

Edwin Vivian El-Kareh, *Secretary and Editor*

Wayne Catlin
Alec Dara-Abrams
Lee Eccles
Fernando Gen-Kuong
Rob A. Hamilton
Robert Johnson
John Lawson
Kang Lee
Tremont Miao
David Moberly
James O. Moore
David E. Rasmussen
Bob Stricklin

Other individuals who contributed to this standard are:

Lee Barford
Janusz Bryzek
Steven Chen
Milton Cram
Jeff Cranmer
John C. Eidson
Robert Gavin
Mike Geipel
Ronald S. Gyurcsik
Bruce Hamilton
John Houldsworth
Kenn Jennyc
Lyle D. Johnsen
Norm LeComte
Michael Mattes
Pradip Madan
John Reidy
Rod Sinks
Hans Sitte
Dale Sogge
David R. Smith

The following persons were on the balloting committee:

William J. Alexander
John Baker
Thomas J. Batko
Dan Carnahan
L. Wayne Catlin
Michael G. Clark
John C. Cole
Rodney Cummings
Walter Czarnocki
Richard T. D'Aquanni
Cyrilla Jane Dalstra
Richard DeMars
Mario D. Dianora
Robert Dolin
Joe Drake
Joseph E. Dryer
David P. Eckel
David Ferguson
Joe Field
David M. Foye
Randy Frank
Carl B. Freidhoff
Mike Geipel
Patrick S. Gonia
Ross S. Gottlieb
Maris Graube
Alfonso C. Guativa
Timothy J. Gundrum
Ronald Gyurcsik
James A. Hammervold
David Howarth
Mutsuya Ii
Daniel LaBonte
Kang Lee
John S. Leffler
Nadim Maluf
Vinod Maudgal
Murray Nicolson
Yukio Nishikawa
J. Mark Noworolski
Adam Pawlikiewicz
Thomas L. Phinney
Josep Samitier
Dale Shoup
Rajiv K. Singh
Grant Smith
Samuel M. Smith
Bob Stricklin
Joseph J. Suter
Arie van Rhijn
Gertjan Van Sprakelaar
Daniel R. Weber
Thomas E. Wheatley
Keith Witney
Stan Woods
James Xu

[1]Numbers preceded by P are IEEE authorized standards projects that were not approved by the IEEE Standards Board at the time this publication went to press. For information about obtaining drafts, contact the IEEE.

Copyright © 1998 IEEE. All rights reserved.

When the IEEE Standards Board approved this standard on 16 September 1997, it had the following membership:

Donald C. Loughry, *Chair* **Richard J. Holleman,** *Vice Chair*
Andrew G. Salem, *Secretary*

Clyde R. Camp
Stephen L. Diamond
Harold E. Epstein
Donald C. Fleckenstein
Jay Forster*
Thomas F. Garrity
Donald N. Heirman
Jim Isaak
Ben C. Johnson
Lowell Johnson
Robert Kennelly
E. G. "Al" Kiener
Joseph L. Koepfinger*
Stephen R. Lambert
Lawrence V. McCall
L. Bruce McClung
Marco W. Migliaro
Louis-François Pau
Gerald H. Peterson
John W. Pope
Jose R. Ramos
Ronald H. Reimer
Ingo Rüsch
John S. Ryan
Chee Kiow Tan
Howard L. Wolfman

*Member Emeritus

Also included are the following nonvoting IEEE Standards Board liaisons:

Satish K. Aggarwal
Alan H. Cookson

Noelle D. Humenick
IEEE Standards Project Editor

Copyright © 1998 IEEE. All rights reserved.

Contents

1. Overview 1

1.1 Scope 2
1.2 Purpose 2
1.3 Conformance 2

2. References 3

3. Definitions, acronyms, and abbreviations 5

3.1 Definitions 5
3.2 Acronyms and abbreviations 8
3.3 Data types 9

4. Smart transducer functional specification 17

4.1 Foundation 17
4.2 Transducer channel types 17
4.3 Functions 19
4.4 Addressing 19
4.5 Interface data transport 23
4.6 Triggering 26
4.7 Control 41
4.8 Status 41
4.9 Interrupt masks 46
4.10 Interrupts 47
4.11 Hot-swap capability 47
4.12 Channel groupings 47
4.13 STIM version 48

5. Transducer Electronic Data Sheet (TEDS) specification 49

5.1 Meta-TEDS data block 49
5.2 Channel TEDS Data Block 59
5.3 Calibration TEDS data block 71
5.4 Meta-identification TEDS data block 80
5.5 Channel Identification TEDS data block 85
5.6 Calibration Identification TEDS data block 89
5.7 End-Users' Application-Specific TEDS data block 92
5.8 Generic Extension TEDS data block 94

6. Transducer Independent Interface (TII) specification 97

6.1 Principles 97
6.2 Line definition 99
6.3 Protocols 99
6.4 Timing 104
6.5 Electrical specifications 109
6.6 Physical specification 113

Annex A (informative) Bibliography 114

Copyright © 1998 IEEE. All rights reserved.

IEEE Standard for a Smart Transducer Interface for Sensors and Actuators—Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats

1. Overview

This standard introduces the concept of a Smart Transducer Interface Module (STIM). A STIM can range in complexity from a single sensor or actuator to many channels of transducers (sensors or actuators). Seven types of transducer channels are recognized by this standard and provision has been made for other types to be added. The transducer channel types are specified in 4.2.

A transducer channel is denoted “smart” in this context because of the following three features:

— It is described by a machine-readable, Transducer Electronic Data Sheet (TEDS).
— The control and data associated with the channel are digital.
— Triggering, status, and control are provided to support the proper functioning of the channel.

A STIM is controlled by a Network Capable Application Processor (NCAP) module by means of a dedicated digital interface. This interface is not a network. The NCAP mediates between the STIM and a digital network, and may provide local intelligence.

This standard is divided into six clauses. Clause 1 provides the scope, the purpose, and the conformance requirements of this standard. Clause 2 lists references to other standards and documents that are useful in applying this standard. Clause 3 provides definitions that are either not found in other standards, or have been modified for use with this standard, as well as a list of acronyms and abbreviations. Clause 4 specifies the functions required of a STIM and of each channel it comprises. Clause 5 specifies the TEDS structure. Clause 6 specifies the physical transducer interface between the STIM and the NCAP.

Copyright © 1998 IEEE. All rights reserved.

1.1 Scope

This standard defines a digital interface for connecting transducers to microprocessors. It describes a TEDS and its data formats. It defines an electrical interface, read and write logic functions to access the TEDS, and a wide variety of transducers. This standard does not specify signal conditioning, signal conversion, or how the TEDS data is used in applications.

1.2 Purpose

There is currently no defined independent digital communication interface standard between transducers and microprocessors. Each vendor builds its own interface. Without an independent, openly defined interface, transducer interfacing and integration are time-consuming and duplicated efforts by vendors are economically unproductive. This interface provides a minimum implementation subset that allows self-identification and configuration of sensors and actuators, and also allows extensibility by vendors to provide growth and product differentiation.

1.3 Conformance

The philosophy underlying the conformance requirements of this subclause is to provide the structure necessary to raise the level of interoperability of transducers and systems built to this standard, while leaving open opportunity for continued technical improvement and differentiation.

A STIM implementation shall be deemed in conformance with this standard provided that the following three requirements are met:

— The STIM supports the required functional performance specified in Clause 4.
— The STIM contains a TEDS that has the format specified in Clause 5.
— The STIM physical interface implements the lines, protocol and timing as defined in Clause 6.

An NCAP implementation shall be deemed in conformance with this standard provided that the following requirement is met:

— An interface that implements the lines, protocols, and timing as defined in Clause 6 is used to access a STIM.

1.3.1 Conformance levels

Several keywords are used to differentiate among various levels of requirements and optionality, as follows.

The word *shall* is used to indicate a mandatory requirement. Designers are required to implement all such mandatory requirements to ensure interoperability with other IEEE 1451.2 conformant products.

The word *recommended* is used to indicate flexibility of choice with a strong preference alternative. The word *should* has the same meaning.

The word *may* is used to indicate a course of action permissible within the limits of the standard, but with no implied preference.

 Copyright © 1998 IEEE. All rights reserved.

2. References

This standard shall be used in conjunction with the following publications. When the following standards are superseded by an approved revision, the revision shall apply.

ANSI X3.4-1986 (Reaff 1992), Coded Character Set—7-bit American National Standard Code For Information Interchange.[1]

CNS 11643: 1992, Standard Interchange Code for Generally-Used Chinese Characters.[2]

FSS-UTF, File System Safe UCS Transformation Format (FSS_UTF). X/Open CAE Specification C501 ISBN 1-85912-082.[3]

GB 2312-1980, China State Bureau of Standards. Coded Chinese Graphic Character Set for Information Interchange.

IEEE Std 754-1985 (Reaff 1990), IEEE Standard for Binary Floating-Point Arithmetic (ANSI).[4]

IEEE Std 1003.1b-1993, IEEE Standard for Information Technology—Portable Operating System Interfaces (POSIX®).

ISO 639: 1988, Code for the Representation of Names of Languages.[5]

ISO 8859-1: 1987, Information Processing—8-Bit Single-Byte Coded Graphic Character Sets—Part 1: Latin Alphabet No. 1.

ISO 8859-2: 1987, Information Processing—8-Bit Single-Byte Coded Graphic Character Sets—Part 2: Latin Alphabet No. 2.

ISO 8859-3: 1988, Information Processing—8-Bit Single-Byte Coded Graphic Character Sets—Part 3: Latin Alphabet No. 3.

ISO 8859-4: 1988, Information Processing—8-Bit Single-Byte Coded Graphic Character Sets—Part 4: Latin Alphabet No. 4.

ISO 8859-6: 1987, Information Processing—8-Bit Single-Byte Coded Graphic Character Sets—Part 6: Latin/Arabic Alphabet.

ISO 8859-7: 1987, Information Processing—8-Bit Single-Byte Coded Graphic Character Sets—Part 7: Latin/Greek alphabet.

ISO 8859-8: 1988, Information Processing—8-Bit Single-Byte Coded Graphic Character Sets—Part 8: Latin/Hebrew alphabet.

---

[1]ANSI publications are available from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA.

[2]CSBS documents can be obtained from the China State Bureau of Standards, P.O. Box 8010, 42 Hi Chun Road, Haidian District, Beijing 100088, China.

[3]This document is available by contacting X/Open Company Ltd—USA, 3141 Fairview Park Drive, Suite 670, Falls Church VA 22042-4501 USA.

[4]IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA.

[5]ISO and ISO/IEC publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse. ISO publications are also available in the United States from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA.

Copyright © 1998 IEEE. All rights reserved.

ISO/IEC 646: 1991, Information Technology—ISO 7-Bit Coded Character Set For Information Interchange.

ISO/IEC 2022: 1994, Information Processing—ISO 7-Bit and 8-Bit Coded Character Sets: Code Extension Techniques.

ISO/IEC 6429: 1992, Information Technology—Control Functions For Coded Character Sets.

ISO/IEC 8859-5: 1988, Information Processing—8-Bit Single-Byte Coded Graphic Character Sets—Part 5: Latin/Cyrillic alphabet.

ISO/IEC DIS 8859-5, Information Technology—8-Bit Single-Byte Coded Graphic Character Sets—Part 5: Latin/Cyrillic Alphabet (Revision of ISO/IEC 8859-5:1988).[6]

ISO/IEC DIS 8859-6: 1987, Information Technology—8-Bit Single-Byte Coded Graphic Character Sets—Part 6: Latin/Arabic Alphabet (Revision of ISO 8859-6: 1987).

ISO/IEC DIS 8859-8, Information Technology—8-Bit Single-Byte Coded Graphic Character Sets—Part 8: Latin/Hebrew Alphabet (Revision of ISO 8859-8:1988).

ISO/IEC 8859-9: 1989, Information Processing—8-Bit Single-Byte Coded Graphic Character Sets—Part 9: Latin Alphabet No. 5.

ISO/IEC DIS 8859-9, Information Technology—8-Bit Single-Byte Coded Graphic Character Sets—Part 9: Latin Alphabet No. 5 (Revision of ISO/IEC 8859-9:1989).

ISO/IEC 8859-10: 1992, Information Technology—8-Bit Single-Byte Coded Graphic Character Sets—Part 10: Latin Alphabet No. 6.

ISO/IEC DIS 8859-10, Information Technology—8-Bit Single-Byte Coded Graphic

ISO/IEC 10646-1: 1993, Information Technology—Universal Multiple-Octet Coded Character Set (UCS)—Part 1: Architecture and Basic Multilingual Plane.

ISO/IEC 10646-2: 1993, Information Technology—Universal Multiple-Octet Coded Character Set (UCS)—Part 2: Ideographic Character Sets.

JIS X 0208-1990, Japanese Standards Association. Code of the Japanese Graphic Character Set for Information Interchange.[7]

JIS X 0212-1990, Japanese Standards Association. Code of the Supplementary Japanese Graphic Character Set for Information Interchange.

KS C 5601-1987, Korea Bureau of Standards. Korean Graphic Character Set for Information Interchange.[8]

TIS 620-2533: 1990, Thai Character Codes for Computers.[9]

[6]Presently at state of Draft International Standard.

[7]JSA documents can be obtained from the Japanese Standards Association, 1-24, Akasaka 4-chome, Minato-Ku, Tokyo, Japan 107.

[8]KBS documents can be obtained from the Korean Bureau of Standards, 2 Chung-ang-dong Kwach'on-city, Kyonggi-do 171-11, Republic of Korea.

[9]TIS documents can be obtained from the Thai Industrial Standards Institute, Rama 6 Street, Bangokok 10400, Tel.202-3348, Fax 247-8739.

 Copyright © 1998 IEEE. All rights reserved.

3. Definitions, acronyms, and abbreviations

This clause provides definitions that are either not found in other standards or have been modified for use with this standard. Also included in this clause are definitions of data types, string language codes, compact physical units, and a universal unique identification system. A list of acronyms and abbreviations can be found in 3.2.

3.1 Definitions

This subclause contains terms as they are used in this standard.

3.1.1 acknowledgment: A signal that is used to reply to a message or signal originator that its message or signal was received.

3.1.2 actuator: A transducer that accepts an electrical signal and converts it into a physical action.

3.1.3 address: A character or group of characters that identifies a register, a particular part of storage, or some other data source or destination. This standard uses *functional addresses* and *channel addresses* to control the flow of data and configuration information between the Network Capable Application Processor and the Smart Transducer Interface Module.

3.1.4 analog to digital converter: A circuit whose input is information in analog form and whose output is the same information in digital form.

3.1.5 buffer: An intermediate data storage location used to compensate for the difference in rate of flow of data or time of occurrence of events when transmitting information from one device to another.

3.1.6 buffered channel: A channel in which the data is placed into a buffer prior to a trigger event and then transmitted or acted upon following that trigger event. This contrasts with an unbuffered channel in which the data is not taken by, or available to, the channel until following the trigger event.

3.1.7 byte: Eight bits of data. Synonym: octet

3.1.8 calibration: The determination of the data to reside in the Calibration Transducer Electronic Data Sheet and to be used for correction.

3.1.9 channel: A single flow path for digital data or an analog signal, usually in distinction from other parallel paths. An IEEE 1451.2 channel provides a path for a single commodity or logical state, either real or virtual, using a single data model and a single set of physical units.

3.1.10 channel address: The portion of a full data transport address that specifies the channel to which the read or write is directed.

3.1.11 channel groupings: Channel groups are manufacturer specifications that define the inherent relationships between the channels of a multichannel Smart Transducer Interface Module. This grouping information is not normally used by the Smart Transducer Interface Module itself. This information will normally be used by Network Capable Application Processor applications to properly compose human readable displays or in formulating other computations. For example, channel groupings can be used to indicate which channels represent the three vector axes of a three-axis vector measurement.

3.1.12 correction: The evaluation of a multinomial function using information from the Calibration Transducer Electronic Data Sheet together with data from one or more channels.

Copyright © 1998 IEEE. All rights reserved.

3.1.13 data conversion: The translation of data from one numeric form into another (e.g., converting a digital-to-analog converter input bit stream into a voltage).

3.1.14 data model: The numeric format in which the Smart Transducer Interface Module will output or accept data.

3.1.15 data sequence sensor: A sensor that samples data independent of any triggers from a network capable application processor.

3.1.16 data sheet: A set of information on a device that defines the parameters of operation and conditions of usage (usually produced by the device's manufacturer).

3.1.17 data structure: A group of digital data fields organized in some logical order for some specific purpose. A two-dimensional paper version of a data structure is an empty fill-in-the-blanks form or an empty tabular chart with organized column and row headings. A data structure is the template by which data is stored in computer memory.

3.1.18 digital interface: A set of wires and a protocol for transferring information by binary means only.

3.1.19 digital to analog converter: A circuit that converts an input number sequence (digital) into a function of a continuous variable (analog).

3.1.20 electronic data sheet: A data sheet stored in some form of electronically readable memory (as opposed to a piece of paper).

3.1.21 enumeration: The listing of the meaning associated with each binary numeric value possible in a data field's storage. Binary numbers are usually expressed in decimal terms for human convenience. Not all possible numeric values need have a specific meaning. Values without meaning are declared to be unused or reserved for future use. Enumeration is the process of declaring the encoding of human interpretable information in a manner convenient for digital electronic machine storage and interchange. The subclause that defines each transducer electronic data sheet data field that is of data type *enumeration* shall contain a table that defines the meaning of the data field for each binary number possible. The meanings encoded in each data field shall be specific and unique to that data field and only that data field. The value becomes meaningless if not associated with the data field and its defining table.

3.1.22 event sequence sensor: A sensor that detects a change of state in the physical world. The instant in time of the change of state, not the state value, is the "measurement."

3.1.23 full data transport address: The combination of a *functional address* and a *channel address*, that specifies whether data is being read or written, to which function, and to which channel.

3.1.24 functional address: The portion of a full data transport address that specifies the read or write function that is to be performed.

3.1.25 hot swap: The act of connecting or disconnecting a Smart Transducer Interface Module and a Network Capable Application Processor without first turning off the power that the Network Capable Application Processor supplies to the Smart Transducer Interface Module over the Transducer Independent Interface.

3.1.26 interrupt: A signal for a processor to suspend one process and begin another. As implemented in IEEE Std 1451.2-1997, an interrupt is a signal from the Smart Transducer Interface Module that enables it to request service from the Network Capable Application Processor.

 Copyright © 1998 IEEE. All rights reserved.

3.1.27 least significant bit: The bit in the binary notation of a number that is the coefficient of the lowest exponent possible.

3.1.28 meta-: A Greek prefix meaning that which pertains to the whole or overall entity or that which is in common or shared with all member entities comprising the whole.

3.1.29 Meta-TEDS: The collection of those Transducer Electronic Data Sheet data fields that pertain to the whole or overall entity or those that are in common or shared with all member entities (channels) comprising the whole transducer product.

3.1.30 most significant bit: The bit in the binary notation of a number that is the coefficient of the highest exponent possible.

3.1.31 multinomial: A linear sum of terms involving powers of more than one variable.

$$\sum_{i_1=0}^{N_1}\sum_{i_2=0}^{N_2}\cdots\sum_{i_m=0}^{N_m} A(i_1, i_2, \ldots i_m)X_1^{i_1}X_2^{i_2}\ldots X_m^{i_m}$$

3.1.32 negative logic: An electronic logic system where the voltage representing one, active, or true has a more negative value than the voltage representing zero, inactive, or false. Also known as negative-true logic, it is normally used in electronic and computing data and communications switching systems for noise immunity reasons.

3.1.33 Network Capable Application Processor: A device between the Smart Transducer Interface Module and the network that performs network communications, Smart Transducer Interface Module communications, and data conversion functions.

3.1.34 not a number: As defined in IEEE Std 754-1985, a bit pattern of a single or double precision real number data type that is a result of an invalid floating point operation.

3.1.35 octet: A group of 8 bits, also known as a byte.

3.1.36 pacing: A method to regulate the flow of bytes read from or written to a Smart Transducer Interface Module.

3.1.37 positive logic: An electronic logic system where the voltage representing one, active, or true has a more positive value than the voltage representing zero, inactive, or false. It is normally used in industrial and commercial control switching systems for safety reasons.

3.1.38 read frame: The transfer of data from a Smart Transducer Interface Module to a Network Capable Application Processor.

3.1.39 sensor: A transducer that converts a physical, biological, or chemical parameter into an electrical signal.

3.1.40 setup time: The period of time during which a system or component is being prepared for a specific operation.

3.1.41 signal conditioning: Sensor signal processing involving operations such as amplification, compensation, filtering, and normalization.

3.1.42 smart actuator: An actuator version of a smart transducer.

Copyright © 1998 IEEE. All rights reserved.

3.1.43 smart sensor: A sensor version of a smart transducer.

3.1.44 smart transducer: A transducer that provides functions beyond those necessary for generating a correct representation of a sensed or controlled quantity. This functionality typically simplifies the integration of the transducer into applications in a networked environment.

3.1.45 Smart Transducer Interface Module: A module that contains the Transducer Electronic Data Sheet, logic to implement the transducer interface, the transducer(s) and any signal conversion or signal conditioning. This standard expressly requires that no operating mode of the Smart Transducer Interface Module ever permit these components to be physically separated. They may, however, be separated during manufacturing and repair.

3.1.46 transducer: A device converting energy from one domain into another. The device may either be a sensor or an actuator.

3.1.47 Transducer Electronic Data Sheet: A data sheet describing a transducer stored in some form of electronically readable memory.

3.1.48 Transducer Independent Interface: The digital interface used to connect a Smart Transducer Interface Module to a Network Capable Application Processor.

3.1.49 transducer interface: The physical connection by which a transducer communicates with the control or data systems that it is a member of, including the physical connector, the signal wires used and the rules by which information is passed across the connection.

3.1.50 transfer: To transmit, or copy, information from one device to another.

3.1.51 trigger: A signal to start an action. As defined in IEEE Std 1451.2-1997 a trigger is a signal from the Network Capable Application Processor serving as a command to the Smart Transducer Interface Module for an action to occur.

3.1.52 trigger cycle: A complete cycle comprising the assertion of the trigger signal by the Network Capable Application Processor followed by the acknowledgment by the Smart Transducer Interface Module.

3.1.53 virtual channel: A channel that behaves as a transducer from the point of view of the Network Capable Application Processor even though nothing outside of the Smart Transducer Interface Module is sensed or changed. Virtual channels are useful for setting or reading operating parameters of other channels.

3.1.54 write frame: The transfer of data from a Network Capable Application Processor to a Smart Transducer Interface Module.

3.2 Acronyms and abbreviations

This subclause contains acronyms and abbreviations of key terms as they are used in this standard.

ack	acknowledgment
ADC	analog-to-digital converter
ASIC	application-specific integrated circuit
DAC	digital-to-analog converter
DI/O	digital input/output
EEPROM	electrically-erasable programmable read-only memory

 Copyright © 1998 IEEE. All rights reserved.

FPGA	field-programmable gate array
ID	identification
lsb	least significant bit
msb	most significant bit
NaN	not a number
NCAP	Network Capable Application Processor
r/w	read/write
SI	International System of Units
STIM	Smart Transducer Interface Module
TEDS	Transducer Electronic Data Sheet
TII	Transducer Independent Interface
UTC	Universal Coordinated Time
xdcr	transducer

3.3 Data types

3.3.1 Unsigned byte integer

3.3.1.1 Usage for counting

Symbol: U8C

Size: 1 byte

This data type is a byte that represents the positive counting integers from zero to 255.

3.3.1.2 Usage for enumeration

Symbol: U8E

Size: 1 byte

All 1 byte enumerations are encoded as unsigned byte integers, with a one to one correspondence between the meaning for a decimal number between zero and 255 (as shown in the respective subclause's defining table) and the actual binary encoding using the unsigned byte integer format in the memory device.

3.3.1.3 Usage for field length

Symbol: U8L

Size: 1 byte

All 1 byte lengths are encoded as unsigned byte integers with the decimal value corresponding to the encoded binary number being the length in bytes of the specified field or data block. In the case of character fields, the length shall be the number of bytes and not the number of characters, since more than one byte may be needed to encode a character. The interpretation of the character strings will need the number of bytes per character to determine the length in characters of a particular string.

Copyright © 1998 IEEE. All rights reserved.

When used to specify the length of a data block (more than one field), the length field shall not include the length of the length field itself.

3.3.2 Unsigned 16 bit integer

3.3.2.1 Usage for counting

Symbol: U16C

Size: 2 bytes

This data type is sixteen bits (2 bytes) that represent the positive counting integers from zero to 65 535.

3.3.2.2 Usage for enumeration

Symbol: U16E

Size: 2 bytes

All 2 byte enumerations are encoded as unsigned integers, with a one to one correspondence between the meaning for a decimal number between zero and 65 535 (as shown in the respective subclause's defining table) and the actual binary encoding using integer format in the memory device.

3.3.2.3 Usage for field length

Symbol: U16L

Size: 2 bytes

All 2 byte lengths are encoded as unsigned integers with the decimal value corresponding to the encoded binary number being the length in bytes of the specified field or data block. In the case of character fields, the length shall be the number of bytes and not the number of characters, since more than one byte may be needed to encode a character. The interpretation of the character strings will need the number of bytes per character to determine the length in characters of a particular string.

When used to specify the length of a data block (more than one field) the length field shall not include the length of the length field itself.

3.3.3 Unsigned 32 bit integer

3.3.3.1 Usage for counting

Symbol: U32C

Size: 4 bytes

Thirty-two bits (4 bytes) that represent the positive counting integers from zero to 4 294 967 295.

 Copyright © 1998 IEEE. All rights reserved.

3.3.3.2 Usage for field length

Symbol: U32L

Size: 4 bytes

All 4 byte lengths are encoded as 32 bit unsigned integers with the decimal value corresponding to the encoded binary number being the length in bytes of the specified field or data block. In the case of character fields, the length shall be the number of bytes and not the number of characters, since more than one byte may be needed to encode a character. The interpretation of the character strings will need the number of bytes per character to determine the length in characters of a particular string.

When used to specify the length of a data block (more than one field), the length field shall not include the length of the length field itself.

3.3.4 Single-precision real

Symbol: F32

Size: 4 bytes

A single-precision real number is a 32 bit (4 bytes) binary sequence that encodes real numbers as specified in IEEE Std 754-1985[10].

3.3.5 Double precision real

Symbol: F64

Size: 8 bytes

A double precision real number is a 64 bit (8 bytes) binary sequence that encodes real numbers as specified in the IEEE Std 754-1985.

3.3.6 String

Symbol: STRING

Size: variable

A string is a consecutive sequence of characters. Each character shall be represented by one or more bytes depending on the encoding scheme specified in 3.3.7. In the TEDS, each string shall be preceded by a length field that declares the length in bytes and not in characters. A string is allowed to contain null bytes (00 hex) in any position or positions. These null bytes shall not be regarded as string terminators. Any null bytes shall be included in the byte count specified in the string length field. A string is data that is not usually processed by the NCAP except for local display.

The most significant byte of a string is the most significant byte of the most significant character encoded by the specified Character Code Format. The most significant character is the first character of the string as it would be written by a human using the language specified in 3.3.7.

[10]Information on references can be found in Clause 2.

Copyright © 1998 IEEE. All rights reserved.

3.3.7 String language specification

Symbol: LANG

Size: 3 bytes

A binary sequence of 3 bytes that encode the String Character Set, Character Code Format, And String Language Code as shown in Table 1.

Table 1—String language fields

Field no.	Field name	No. of bytes
1	String Character Set	1
2	Character Code Format	1
3	String Language Code	1

3.3.7.1 String Character Set

Data type: unsigned byte integer used for enumeration (U8E, 1 byte).

This field gives the specific character set used to interpret the octet streams after parsing, using the method indicated by the Character Code Format. Table 2 defines values for each String Character Set.

Table 2—Enumeration of String Character Set Codes

Value	Code description standard	Meaning (informative)
0	ISO 8859-1: 1987	Latin set 1
1	ISO 8859-2: 1987	Latin set 2
2	ISO 8859-3: 1988	Latin set 3
3	ISO 8859-4: 1988	Latin set 4
4	ISO/IEC 8859-5: 1988	Cyrillic
5	ISO 8859-6: 1987	Arabic
6	ISO 8859-7: 1987	Greek
7	ISO 8859-8: 1988	Hebrew
8	ISO/IEC 8859-9: 1990	Latin set 5
9	ISO/IEC 8859-10: 1992	Latin set 6
10	ANSI X3.4-1986(Reaff. 1992)	Same as ASCII and ISO/IEC 646: 1991
11	ISO 10646-1: 1993, UCS 2	—
12	ISO 10646-2: 1993, UCS 2	—
13	ISO 10646-1: 1993, UCS 4	—
14	ISO 10646-2: 1993, UCS 4	—
15	JIS X 0208: 1990	Japanese circa 1978
16	JIS X 0212: 1990	Japanese circa 1990
17	KS C 5601-1992	Korea
18	GB 2312-1980	People’s Republic of China
19	CNS 11643: 1992	Taiwan
20	TIS 620-2533: 1990	Thailand
21–255	Reserved for future expansion	—

 Copyright © 1998 IEEE. All rights reserved.

ISO 10646-1993 is listed four times because it has two levels of conformance and two word sizes. The levels of conformance are:

Level 1—No combining characters are allowed
Level 2—Combining characters are allowed

The possible ISO 10646 word sizes are:

UCS 2—Universal code set with 2 octets per character
UCS 4—Universal code set with 4 octets per character

NOTES

1—Combining characters in ISO 10646 are characters that embellish the previous character. For example, a “u” with an umlaut (two dots over it) can be represented two ways: as a single character code having a glyph of a “u” with two dots over it or as two adjacent character codes, the first of which is a glyph of the letter “u,” and the second is a glyph of an umlaut. The second character (the umlaut glyph) is the combining character. With level 1 conformance, string comparisons are easy: just compare the character codes. With level 2 conformance, string comparisons must be done more carefully because the same “character” can have more than one character code representation.

2—Unicode has essentially turned into enumeration 12—ISO 10646: 1993, level 2, UCS 2.

3.3.7.2 Character Code Format

Data type: unsigned byte integer used for enumeration (U8E, 1 byte)

The format, order, direction of display, etc., by which the Character Set Codes are interpreted. The format listed in this field also includes the number of octets making up each character in the string field. The first group of three enumerations are fixed-width formats (a fixed number of bits per character), while the remaining are either variable-width formats (a varying number of bits per character), or code switching formats (the glyph associated with a given code depends on the current mode). Table 3 defines values for each Character Code Format.

Table 3—Enumeration of Character Code Formats

Value	Character Code Formats standard	Meaning (informative)
0	1 Octet fixed-width	—
1	2 Octets fixed-width	—
2	4 Octets fixed-width	—
3	ISO/IEC 2022: 1994	—
4	Shift-JIS (SJIS)	Japan
5	ISO/IEC 6429: 1992	—
6	FSS-UTF: 1995	AT&T File System Safe Universal Transformation Format
7	JIS X 0208: 1990	Extended UNIX Code (EUC)
8–255	Reserved for future expansion	—

Copyright © 1998 IEEE. All rights reserved.

3.3.7.3 String Language Code

Data type: unsigned byte integer used for enumeration (U8E, 1 byte)

This field indicates which language the string fields in the TEDS are written in. This is used in conjunction with the String Character Set because many character sets support more than one language. For example, the ISO 8859-1: 1987 character encoding supports English, German, French, Danish, Dutch, etc.

The enumerated value corresponds to the numeric sequence of the language in the alphabetic list of two-letter language symbols in ISO 639: 1988. Table 4 lists some of the possible languages. All languages listed in ISO 639: 1988 have enumerated values. The enumerated value for languages not listed in Table 4 can be found by consulting the alphabetic list of two-letter language symbols in ISO 639: 1988. Languages and dialects not listed in ISO 639: 1988 shall not use the reserved enumeration values from 137 to 255 and shall not be used in the standard-defined TEDS string fields.

Table 4—Enumeration of String Language Codes

Value	ISO 639 language code	Meaning (informative)
0	Reserved	—
1	aa	Afar
21	da	Danish
22	de	German
25	en	English
27	es	Spanish
29	eu	Basque
31	fi	Finnish
34	fr	French
36	ga	Irish
51	it	Italian
82	nl	Dutch
83	no	Norwegian
88	pl	Polish
90	pt	Portuguese
95	ru	Russian
112	sv	Swedish
130	vi	Vietnamese
136	zu	Zulu
137–255	Reserved	—

 Copyright © 1998 IEEE. All rights reserved.

3.3.8 Physical units

Symbol: UNITS

Size: 10 bytes

A binary sequence of 10 bytes that encodes physical units is described in Table 5. Each of the fields shall be interpreted as an unsigned integer. A unit shall be represented as a product of the SI base units, plus radians and steradians, each raised to a rational power. This structure encodes only the exponents; the product is implicit.

Table 5—Physical units data type structure

Field no.	Description	No. of bytes
1	ENUMERATION 0: Unit is described by the product of SI base units, plus radians and steradians, raised to the powers recorded in fields 2–10. 1: Unit is U/U, where U is described by the product of SI base units, plus radians and steradians, raised to the powers recorded in fields 2–10. 2: Unit is $\log_{10}$(U), where U is described by the product of SI base units, plus radians and steradians, raised to the powers recorded in fields 2–10. 3: Unit is $\log_{10}$(U/U), where U is described by the product of SI base units, plus radians and steradians, raised to the powers recorded in fields 2–10. 4: The associated quantity is digital data (e.g., a bit vector) and has no unit. Fields 2–10 shall be set to 128. The “digital data” type applies to data that represents counting, such as the output of an ADC, or that represents a state, such as the current positions of a bank of switches. 5: The associated physical quantity is represented by values on an arbitrary scale (e.g., hardness). Fields 2–10 are reserved, and shall be set to 128. 6–255: Reserved	1
2	(2 * <exponent of radians>) + 128	1
3	(2 * <exponent of steradians>) + 128	1
4	(2 * <exponent of meters>) + 128	1
5	(2 * <exponent of kilograms>) + 128	1
6	(2 * <exponent of seconds>) + 128	1
7	(2 * <exponent of amperes>) + 128	1
8	(2 * <exponent of kelvins>) + 128	1
9	(2 * <exponent of moles>) + 128	1
10	(2 * <exponent of candelas>) + 128	1

The U/U forms (enumerations 1 and 3) are for expressing “dimensionless” units such as strain (meters per meter) and concentration (moles per mole). The numerator and denominator units are identical, each being specified by subfields 2–10.

Boolean data (values in {0, 1} or {False, True}) shall be represented as digital data (enumeration 4) with Channel Data Model Length = 1 and Channel Model Significant Bits = 1. See 5.2.3.14 through 5.2.3.16.

For further information see [B1].[11]

[11]The numbers in brackets correspond to those of the bibliography in Annex A.

Copyright © 1998 IEEE. All rights reserved.

3.3.9 Universal unique identification

Symbol: UUID

Size: 10 bytes

The UUID is an identification field associated with the STIM whose value is unique in the universe. There shall be no requirement that the interpretation of the UUID reflect the actual place or time of manufacture of the STIM. The use of time and location in the algorithm shall be used only to ensure uniqueness. Thus, a manufacturer shall be allowed to use the fields defined by the algorithm as desired, provided that when interpreted as defined in Table 6, there shall be no other possible manufacturer that could claim use of the same UUID.

The UUID field shall be 10 bytes long and consist of four subfields (from most to least significant order: location, manufacturer, year, and time) as defined in Table 6.

Table 6—Universal unique identification data type structure

Field no.	Description	No. of bits
1	Location field: The value of this field shall be chosen by the STIM manufacturer to identify a particular location on the earth, the "location," over which the manufacturer has physical control. This value may represent the actual location of STIM manufacture. A manufacturer may use different values of this field in the operation as long as all values meet the requirements of this subclause. —The representation of the location field shall be 42 bits. —The msb shall indicate North (asserted) or South (not asserted) latitude. —The next 20 msbs of this field represent the magnitude of the "location" latitude as an integer number of arc seconds. —The next most significant bit shall indicate East (asserted) or West (not asserted) longitude. —The remaining 20 bits shall represent the magnitude of the "location" longitude as an integer number of arc seconds. Latitude magnitude values greater than 90° are reserved. Longitude magnitude values of greater than 180° are reserved. NOTE—One arc second at the equator is about 30 m. Thus the range represented by each 20 bit field is zero to 1 048 575 arc seconds or 0–291°, which is sufficient to represent latitude and longitude on the earth's surface.	42
2	Manufacturer's field: The value of this field may be chosen by the STIM manufacturer for any purpose provided there are no interferences in the use of the location field. A location field interference shall mean that there is more than one manufacturer that could claim physical control over the location defined by the location field. If such interferences exist, all interfering manufacturers shall negotiate the use of the manufacturer's field values to resolve any interferences, (i.e., the combination of the location field and the manufacturer's fields shall unambiguously define a specific STIM manufacturer.) This negotiation shall be reopened every time there is a change in the interference.	4
3	Year field: The value of this field shall be the value of the current year. The representation of the year field shall be a 12 bit integer value. The range of this field shall be the years 0 to 4095 AD. The beginning of the year shall be deemed to start on January 1, 00:00:00, UTC.	12
4	Time field: This value shall be chosen by the STIM manufacturer such that, in combination with the location, manufacturer, and year fields, the resulting UUID is unique for all STIMs made under the manufacturer's control. The choice of values for the time field shall be further restricted such that the values, when interpreted as the time since the beginning of the year, do not represent times prior to the manufacturer obtaining physical control over the "location" nor values in the future in which the manufacturer cannot guarantee physical control over the "location." The representation of the time field shall be a 22b integer. The range shall be 0 to 4 194 303. When it is necessary to interpret this field as time since the beginning of the year, the representation shall be an integer number of 10 s intervals. In this case, time values greater than one year are reserved. NOTE—There are approximately 31 536 000 seconds per year.) The beginning of the year shall be deemed to start on January 1, 00:00:00, UTC.	22

 Copyright © 1998 IEEE. All rights reserved.

4. Smart transducer functional specification

This clause specifies the types of transducer channels that may be implemented in a STIM, and the functions required of the STIM to permit channels to work together. It further defines addressing, data transport, triggering, control, and status functions.

4.1 Foundation

The context for the transducer interface specification is shown in Figure 1. A STIM may be used to sense or control multiple physical phenomena. Each phenomenon sensed or controlled shall be associated with a STIM transducer channel. A channel may be a virtual transducer in the sense that it behaves as a sensor or actuator, even though nothing outside of the STIM is sensed or changed.

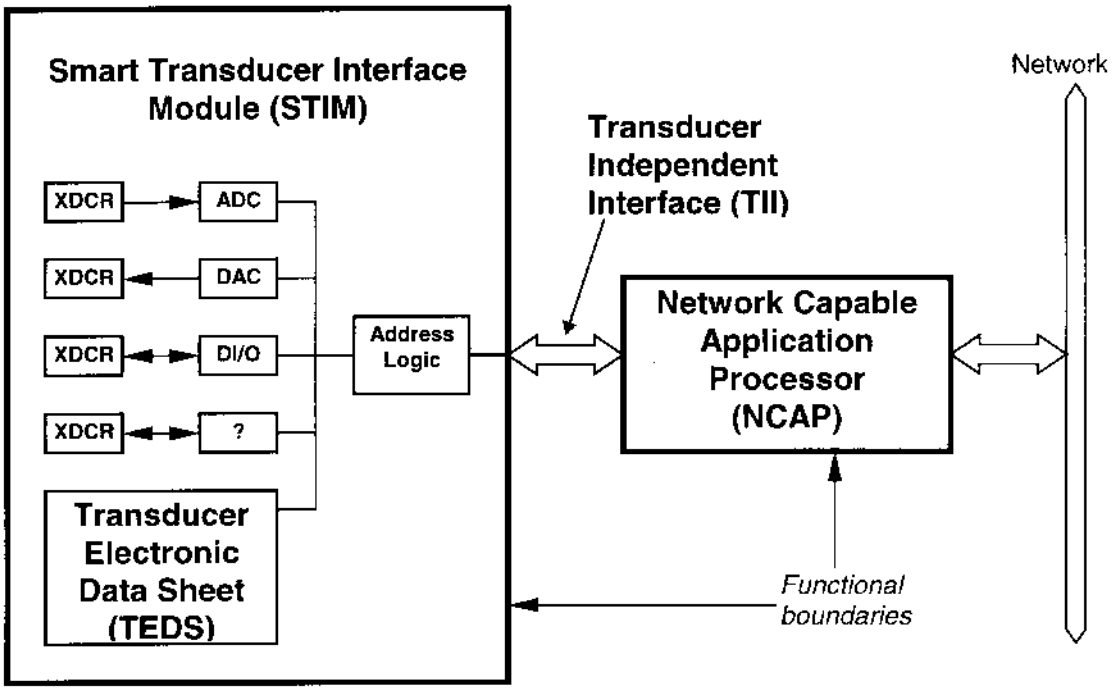

Figure 1—Context for the transducer interface specification

4.2 Transducer channel types

This subclause specifies the general behavior of six channel types. An additional seventh channel type is identified to allow for extensions to STIM behavior beyond those specified in this subclause. The detailed timing and control of these channel types are specified in later subclauses of this standard. The seven channel types are as follows:

— Sensor
— Actuator
— Buffered sensor
— Data sequence sensor
— Buffered data sequence sensor
— Event sequence sensor
— General transducer

Copyright © 1998 IEEE. All rights reserved.

4.2.1 Sensor

A sensor measures some physical parameter on demand and returns digital data representing that parameter. A new data set shall be sampled once for each triggering event. The data set available to be read shall be the data set acquired as a result of the most recent trigger event. The number of data points in the data set shall be determined by the Channel Data Repetitions field (see 5.2.3.17).

4.2.2 Actuator

An actuator causes a physical or virtual action to occur that shall be related to the data set sent to the actuator. The actuator state changes to match the data set most recently sent to it when a triggering event occurs. The number of data points in the data set shall be determined by the Channel Data Repetitions field (see 5.2.3.17).

4.2.3 Buffered sensor

A buffered sensor differs from a simple sensor (4.2.1) in that it has a single level of data buffering on the output channel. A new data set shall be sampled once for each triggering event. The data set available to be read shall be the data set acquired on the second most recent trigger event. The number of data points in the data set shall be determined by the Channel Data Repetitions field (see 5.2.3.17).

4.2.4 Data sequence sensor

A data sequence sensor acquires data continuously, with sampling times under control of the STIM. Triggering selects a data set from this stream of continuous measurements and makes it available to be read by the NCAP. The data set selected shall be the one acquired immediately following the trigger. The data collection process in the STIM shall be enabled or disabled by means of a control command as defined in 4.7.

The data set acquisition timing need not be periodic. The number of data points in the data set shall be determined by the Channel Data Repetitions field (see 5.2.3.17).

4.2.5 Buffered data sequence sensor

A buffered data sequence sensor acquires data continuously, with sampling times under control of the STIM. Triggering selects a data set from this stream of continuous measurements and makes it available to be read by the NCAP. The data set selected shall be the one acquired immediately previous to the trigger. The data collection process in the STIM shall be enabled or disabled by means of a control command as defined in 4.7.

The data set acquisition timing need not be periodic. The number of data points in the data set shall be determined by the Channel Data Repetitions field (see 5.2.3.17).

4.2.6 Event sequence sensor

An event sequence sensor produces a signal whenever a specific event occurs. The signal shall be the same signal used by sensors and actuators to acknowledge triggering events. The event may be a digital signal transition or an analog signal crossing a setpoint. An event sequence sensor may be configured to signal an event on high-to-low transitions or low-to-high transitions, or both. Reading and writing to an event sequence sensor channel is used for configuration and state information only. The time of the event is conveyed by the trigger acknowledge signal. Other actuator and sensor channels may be associated with an event sequence sensor to allow changing analog setpoints or hysteresis, or reading the analog value being sensed. This association is communicated to the NCAP through the grouping information in the Meta-TEDS.

 Copyright © 1998 IEEE. All rights reserved.

4.2.7 General transducer

This category of channel allows for the presence of channels that behave differently from the above types. These types shall implement the same functions required of all channels, but this standard does not specify behavior with respect to triggering or data written to or read from such a channel.

4.3 Functions

Each STIM shall implement the following functions:

- Addressing
- Interface data transport
- Meta-TEDS
- Global status
- Global control
- Triggering
- Hot-swap capability
- Interrupt
- Interrupt masking

Each STIM may also implement an optional Meta-identification TEDS.

Each channel of a STIM shall implement the following functions:

- Channel TEDS
- Transducer data
- Status
- Control

Each channel may also implement the following optional functions:

- Calibration TEDS
- Calibration Identification TEDS
- Channel Identification TEDS
- End-User's Application-Specific TEDS
- Generic Extension TEDS
- Self calibration
- Self-test

The interfaces to the End-User's Application-Specific TEDS, Generic Extension TEDS, self calibration, and self-test are defined, but this standard does not specify their form or behavior. Provision is made for adding additional functions not defined by this standard.

4.4 Addressing

Addressing is used in conjunction with the interface data transport. A full address specifies whether data is being read or written, to which function, and to which STIM channel.

Functional and channel addresses are logical addresses. The mapping of functional or channel addresses to physical addresses shall be accomplished within the STIM and is beyond the scope of this specification.

Copyright © 1998 IEEE. All rights reserved.

4.4.1 Address structure

A full address shall be 2 bytes long and structured as shown in Figure 2.

Functional address Most significant byte								Channel address Least significant byte							
r/w	Function code							Channel number							
msb							lsb	msb							lsb

Figure 2—Address layout

For convenience, the most significant byte is called the functional address and the least significant byte is called the channel address. The functional address is subordinate to the channel address, but is transmitted first to make implementation easier.

4.4.2 Channel addresses

Each transducer in a STIM shall be assigned a channel number. A STIM may have up to 255 channels. The number of implemented channels can be determined by reading the Meta-TEDS.

Implemented channels shall be numbered consecutively starting from one. Every channel number between one and the highest implemented channel number shall address an implemented channel.

Channel address zero has special meaning and is referred to throughout this standard as CHANNEL_ZERO.

When CHANNEL_ZERO is used, the function shall refer to the STIM as a whole or to all channels collectively. The word *global* or the prefix *meta-* is used to modify the functional address name.

4.4.3 Functional addresses

4.4.3.1 Read/Write bit

The msb of the functional address is used to specify the direction of data communication over the interface, according to Table 7.

Table 7—Communication direction bit values

Value	Communications direction
0	Write to the STIM
1	Read from the STIM

The remaining bits of the functional address denote the function selected.

Copyright © 1998 IEEE. All rights reserved.

4.4.3.2 Function selected

Tables 8 and 9 specify the most commonly used functional addresses.

Table 8—Commonly used global functional addresses

Address	Function
0	Write global transducer data
1	Write global control command
3	Write triggered channel address
5	Write global standard interrupt mask
96	Write global End-Users' Application-Specific TEDS
128	Read global transducer data
130	Read global standard status
132	Read global auxiliary status
160	Read Meta-TEDS
161	Read Meta-Identification TEDS
224	Read global End-Users' Application-Specific TEDS

Table 9—Commonly used channel functional addresses

Address	Function
0	Write channel transducer data
1	Write channel control command
5	Write channel standard interrupt mask
64	Write Calibration TEDS
65	Write Calibration Identification TEDS
96	Write End-Users' Application-Specific TEDS
128	Read channel transducer data
130	Read channel standard status
132	Read channel auxiliary status
160	Read Channel TEDS
161	Read Channel Identification TEDS
192	Read Calibration TEDS
193	Read Calibration Identification TEDS
224	Read End-Users' Application-Specific TEDS

Table 10 summarizes the functions for all functional addresses.

Copyright © 1998 IEEE. All rights reserved.

Table 10—Functional addresses showing read/write symmetry

Write address	CHANNEL_ZERO function	Channel 1–255 function	Read address	CHANNEL_ZERO function	Channel 1–255 function
Operational addresses					
0	Write global transducer data	Write channel transducer data	128	Read global transducer data	Read channel transducer data
1	Write global control command	Write channel control command	129	Reserved	
2	Reserved		130	Read global standard status	Read channel standard status
3	Write triggered channel address	Reserved	131	Read triggered channel address	Reserved
4	Reserved		132	Read global auxiliary status	Read channel auxiliary status
5	Write global standard interrupt mask	Write channel standard interrupt mask	133	Read global standard interrupt mask	Read channel standard interrupt mask
6	Write global auxiliary interrupt mask	Write channel auxiliary interrupt mask	134	Read global auxiliary interrupt mask	Read channel auxiliary interrupt mask
7	Reserved		135	Read STIM version	Reserved
8–15	Reserved for future write extensions		136–143	Reserved for future read extensions	
16–31	Open for industry write extensions		144–159	Open for industry read extensions	
TEDS addresses					
32	For manufacturer's use only	For manufacturer's use only	160	Read Meta-TEDS	Read Channel TEDS
33	For manufacturer's use only	For manufacturer's use only	161	Read Meta-Identification TEDS	Read Channel Identification TEDS
34–47	Reserved for writing future TEDS extensions		162–175	Reserved for reading future TEDS extensions	
48–63	For manufacturer's use only		176–191	Open for reading industry TEDS extensions	
Calibration TEDS addresses					
64	Reserved	Write Calibration TEDS	192	Reserved	Read Calibration TEDS
65	Reserved	Write Calibration Identification TEDS	193	Reserved	Read Calibration Identification TEDS
66–79	Reserved for writing future Calibration TEDS extensions		194–207	Reserved for reading future Calibration TEDS extensions	
80–95	Open for writing industry Calibration TEDS extensions		208–223	Open for reading industry Calibration TEDS extensions	
General writable storage addresses					
96	Write global End-Users' Application-Specific TEDS	Write channel End-Users' Application-Specific TEDS	224	Read global End-Users' Application-Specific TEDS	Read channel End-Users' Application-Specific TEDS
97–111	Reserved for writing future writable nonvolatile data		225–239	Reserved for reading future writable nonvolatile data	
112–127	Open for writing industry writable nonvolatile data		240–255	Open for reading industry writable nonvolatile data	

Copyright © 1998 IEEE. All rights reserved.

Note that the function has a global interpretation if CHANNEL_ZERO is addressed, and a channel-specific interpretation for channels 1 to 255. Table 10 includes the following four categories of functional addresses:

a) Addresses whose functions are defined completely by this specification
b) Addresses whose functions are defined partially or not at all
c) Addresses that are *reserved* for future versions of the standard
d) Addresses that are *open* for industry use

Functional addresses 0–31 and the corresponding addresses with the read bit set (128–159) are intended for operational functions.

Functional addresses 32–127 and 160–255 deal with data-sheet type information that is primarily read-only and is nonvolatile. Data structures whose addresses fall in these ranges shall have a 4 byte length field at the beginning and a 2 byte checksum at the end. The format of the length field and the checksum is specified in Clause 5.

Functional addresses 32–63 are invalid under normal use since they correspond to TEDS functions 160–191, which are read-only and shall not be modifiable by the user. Functional addresses 32, 33, and 48–63 may be used by a STIM manufacturer to create TEDS via the TII. Manufacturers shall use some mechanism to prevent unauthorized updating of those TEDS. The method of preventing unauthorized modifications is not part of this standard.

Functional addresses in the range of 64–127 allow the possibility of the user changing the data-sheet type structures found at corresponding functional addresses 192–255. For example, addresses 64 and 65 may be used for transducer recalibration.

All functional addresses that are marked as *reserved* are intended for future revisions to this standard and shall not be used by industry for any purpose.

All functional addresses that are marked as *open* may be used by industry. Functional addresses 16–31 and 144–59 may be used to implement extended operational functions. All other functional addresses that are marked as *open* may be used for extensions to the TEDS structures herein defined. In either operational or TEDS extensions, the data structures shall have a 4 byte length field at the beginning and a 2 byte checksum at the end. The format of the length field and the checksum shall be identical to those defined in Clause 5 for TEDS. Clause 5 gives additional restrictions on the data structure associated with TEDS extensions.

It is intended that functional addresses designated as *open* for industry extensions should be defined cooperatively by industry sectors.

The STIM shall respond to any attempt to read or write to any unimplemented or *reserved* functional address by setting the *STIM invalid command* bit in the standard status register. See 4.8 for a complete description of this bit.

4.5 Interface data transport

The data transport shall be supported by a group of signal lines in the physical interface. This service conveys addressing to the STIM and the data associated with the address between the STIM and the NCAP.

4.5.1 Data format

Data shall always consist of an integral number of bytes. When data transport involves multiple byte numeric representations, the most significant byte shall be sent first. For N-byte integer data representation that are not a multiple of 8 bits, pad bits shall be added above the most significant bit to bring the total to a multiple

Copyright © 1998 IEEE. All rights reserved.

of eight. For N-byte fractional data representations that are not a multiple of 8 bits, pad bits shall be added below the least significant bit to bring the total to a multiple of eight.

The physical interface shall transport each data byte in bit-serial form, most significant bit first.

4.5.2 Data transport function

Means shall be provided on the physical interface for the NCAP to signal when the data transport is active, and to delimit data transport frames. Means shall also be provided for the STIM to acknowledge its readiness for data transport.

The data transport function interacts with the trigger function. The data transport function shall be inactivated before the trigger is asserted.

A data transport frame shall begin by the NCAP sending an address to the STIM. The complete address specifies whether data shall be written to or read from the STIM, and which channel and function are involved.

The length of the data structure to be transported shall be determined in one of the following three ways:

a) The lengths of status, interrupt mask, and control data and the triggered channel address are specified in this standard.

b) The length of transducer data shall be determined from the channel data model field and related fields in the channel TEDS.

c) All other data structures shall be preceded by a 4 byte integer indicating the length of the structure excluding the 4 byte length field.

4.5.3 Data transport rate and pacing

Data rates shall be controlled by the NCAP. All STIMs shall support the common data flow rate specified in 6.4. The rate may be changed to a higher rate based on information available in the TEDS.

Means shall be provided for both the STIM and the NCAP to regulate the flow of data bytes within a frame. This is referred to as *pacing*.

4.5.4 Transducer data

Data transport is most frequently used to read data from sensor, buffered sensor, data sequence sensor, and buffered data sequence sensors, and to write data to actuator and event sequence channels. Triggering interacts with data transport of transducer data as specified in 4.6.

Writing data to a sensor, buffered sensor, data sequence sensor, or buffered data sequence sensor shall have no effect.

Reading from a sensor, buffered sensor, data sequence sensor, or buffered data sequence sensor without triggering shall return the same data as when last read.

Reading data from any sensor after an aborted trigger cycle may produce unpredictable results.

Reading data from a buffered sensor or buffered data sequence sensor after initialization or after an aborted trigger cycle may produce unpredictable results until after the second trigger has been sent.

Reading data from an actuator shall return the latest data written to it.

 Copyright © 1998 IEEE. All rights reserved.

Reading data from an actuator after initialization, but before writing data to it, shall return the initial state of the actuator.

The CHANNEL_ZERO functions, *write global transducer data* and *read global transducer data*, result in writing to or reading from the data structures of all implemented channels concatenated together, in order, beginning with channel one. This allows access to all transducers without the addressing overhead associated with accessing each individual channel. There are no delimiters between channel data structures. The data for each channel shall be parsed according to the data models recorded in each Channel TEDS. The behavior of each channel shall conform to the behavior specified for such reads and writes to individual channels.

EXAMPLES

1—A read operation to *read global transducer data* is performed on a STIM that has a sensor on channels 1, 2, and 4 and an actuator on channel 3. The data shall include N_1 bytes for channel 1, N_2 bytes for channel 2, N_3 bytes for channel 3, and N_4 bytes for channel 4 (where N_1 through N_4 are the data model lengths for their respective channels).

2—With mixed sensors and actuators, a write operation to *write global transducer data* is performed on a STIM that has a sensor on channels 1 and 3, and actuators on channels 2 and 4. The data to be written shall be a concatenation of N_1 bytes of dummy data, N_2 bytes of valid data, N_3 bytes of dummy data, and N_4 bytes of valid data (where N_1 through N_4 are the data model lengths for their respective channels). Channels 1 and 3 will be unaffected while channels 2 and 4 will have the new valid data representing the next states of the respective actuators.

4.5.5 Data transport of other functions

Data transport is also used to read data sheet information such as the TEDS and to read status and write control commands to channels. The CHANNEL_ZERO status and control functions are not concatenated channel status and control commands. Instead, global status and control functions are defined.

Some data sheet information is also writable and nonvolatile (Calibration TEDS, End-Users' Application-Specific TEDS). The CHANNEL_ZERO data sheet functions, where defined or allowed, are not the concatenated channel functions. They shall apply to the STIM as a whole.

4.5.6 Data transport exceptions

Data reads and writes are intended to access complete data structures.

Partial reads of data-sheet type structures, that is, functional addresses 160–255 inclusive, are allowed, and shall not leave such functions in a corrupted state. Parsing shall take into account any data misalignment caused by such a partial read.

Partial reads of functional addresses between 128–159 inclusive may leave such functions in a corrupted state.

Partial writes to any functional address may leave the function in a corrupted state.

Reading more bytes than are specified for the data structure accessed shall not cause corruption of state or data for the addressed channel nor any other channel. Writing more bytes than are specified for the data structure accessed may leave the data corrupted for the addressed channel, but shall not cause corruption of state or data for any other channel.

Writing to an unimplemented channel address, or to an invalid functional address of an implemented channel shall not change or corrupt the state of any valid function of any implemented channel. The STIM shall respond to any attempt to address unimplemented channels by setting the *STIM invalid command bit* in the standard status register. See 4.8 for a complete description of this bit.

Copyright © 1998 IEEE. All rights reserved.

4.6 Triggering

Triggering shall be supported by signal lines in the physical interface. The triggering function provides means for an NCAP to send to a STIM a command for an action to take place (the trigger signal), and for the STIM to signal the time when the action occurred (trigger acknowledgment). For example, the action may be that a sensor channel samples new data. Each transducer channel type differs from another chiefly in the way it responds to triggering. The actions of each transducer channel type in response to triggering are described in 4.6.4 through 4.6.9.

Triggering shall only apply to a single channel, or to all channels at once (global triggering). The channel to which the trigger applies is selected by the triggered channel address.

The behavior of the triggering system from the point of view of the STIM is illustrated by the state diagram in Figure 3. The action initiated by normal triggering belongs to a separate, channel-type-dependent, concurrent process. The normal states of this concurrent process and their relationship to the trigger states are shown in Figures 4 through 9. Exceptions brought on by aborted triggering are detailed in the text of 4.6.4 through 4.6.9.

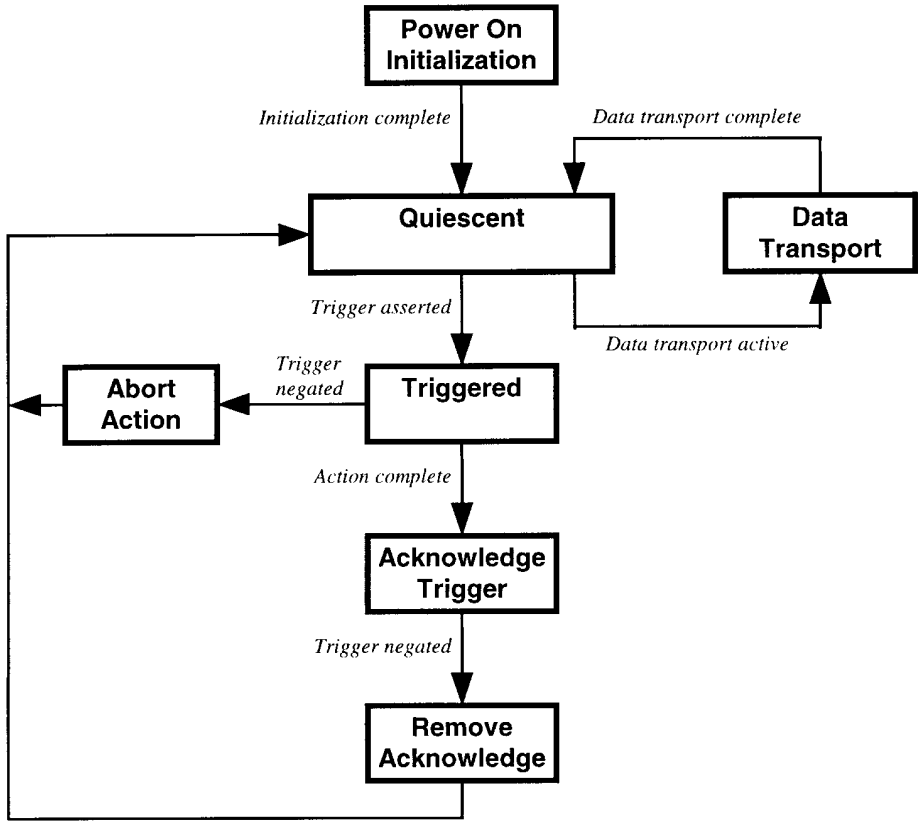

Figure 3—General response of STIM to a trigger

The timing specifications referred to in the transition from invalid to valid data are described further for individual sensor channel types in 4.6.4 through 4.6.9, and for global triggering and reading of multiple transducer types in 4.6.10.

Table 11 shows the STIM states related to triggering.

 Copyright © 1998 IEEE. All rights reserved.

Table 11—STIM states related to triggering

State	Definition
Power On Initialization	The STIM is performing initialization.
Quiescent	The STIM is waiting for either the trigger or the data transport to be active.
Triggered	The STIM has received a trigger and the triggered channel is performing the appropriate actions. (See Figures 4 through 9.)
Abort Action	The NCAP has removed the trigger prematurely to abort the action in the STIM. The resulting transducer data or actuator state is undefined. The state machine proceeds directly to the Quiescent state.
Acknowledge Trigger	The STIM has acknowledged the trigger and is waiting for the NCAP to negate it.
Remove Acknowledge	The NCAP has negated the trigger. The STIM removes the trigger acknowledge. The state machine proceeds directly to the Quiescent state.
Data Transport	The STIM is reading or writing data, status, control, or TEDS data fields. The trigger must remain inactive (negated) until data transport is complete.

4.6.1 The trigger signal

The trigger signal shall be driven by the NCAP. It serves as a command to a STIM for an action to occur. The physical interface shall ensure that the trigger signal is received by the STIM.

The trigger signal shall be negated when the data transport function is active, i.e., when reading from or writing to the STIM is enabled.

The trigger may produce unpredictable results if sent before the write setup time has expired, after a write to the transducer data address. The Channel Write Setup Time is in the Channel TEDS. The Global Worst-Case Write Setup Time, applicable to global triggering, is in the Meta-TEDS.

4.6.2 The trigger acknowledge signal

The trigger acknowledge signal shall be driven by the STIM. It serves as a marker denoting the time an event occurs. In the case of global triggering, trigger acknowledge serves as a reference from which individual event times can be calculated. Trigger acknowledge is also a response to the NCAP confirming that the action requested by the trigger did occur. The timing of trigger acknowledge depends on the transducer type of the triggered channel and is described for each type in 4.6.4 through 4.6.9. See 4.6.10 through 4.6.11 for additional constraints on the behavior of the trigger acknowledge signal.

Reading transducer data from the triggered channel may produce unpredictable results if the read setup time has not expired following trigger acknowledgment. The Channel Read Setup Time is in the Channel TEDS. The Global Worst-Case Read Setup Time, applicable to global triggering, is in the Meta-TEDS.

For virtual transducers or where timing is not important, trigger acknowledge may be sent immediately upon receipt of trigger.

If trigger is asserted and then negated before the trigger is acknowledged, then trigger acknowledge and any remaining action associated with trigger shall be aborted. The state and transducer data of the triggered channel after aborted triggering are undefined until the next completed trigger cycle.

If the trigger acknowledge is not received by the NCAP by the Channel Update Time listed in the channel TEDS (5.2.3.21), or in the case of global triggering by the worst-case Channel Update Time in the Meta-TEDS (5.1.3.13), the NCAP may assume that the triggered channel is not operating properly and abort the trigger by negating the trigger signal.

Copyright © 1998 IEEE. All rights reserved.

All STIM channels shall implement a *trigger acknowledged bit* in the channel standard status register. See 4.8 for the specific behavior of this bit.

4.6.3 The triggered channel address

The triggered channel address specifies the channel to which the trigger applies. If the triggered channel address is within the range of implemented channels, then all triggering shall be directed at that channel alone. If the triggered channel address is CHANNEL_ZERO, then triggering shall be directed at all implemented channels. This is referred to as global triggering.

To change the triggered channel address, the NCAP shall write 1 byte to functional address *write triggered channel address* at CHANNEL_ZERO. The byte value shall be the new triggered channel until changed by another such write. Reading *read triggered channel address* at CHANNEL_ZERO shall return the value of the last byte written to *write triggered channel address*.

There is no specified default triggered channel address upon power-up.

4.6.4 Triggering sensors

The trigger signal shall cause a sensor channel to acquire new data or a new data set. For the simplest sensors, this could mean a digital latch is updated or an analog-to-digital converter begins a conversion. More complex sensors may take multiple samples per trigger, spaced apart incrementally in time or some other dimension. See 4.6.11 for further details.

For a sensor channel, the STIM shall send a trigger acknowledgment coincident with the sample time. Subsequent to this trigger acknowledge and the additional duration specified by the Channel Read Setup Time, the data shall be available to the NCAP. The Channel Read Setup Time is specified in the Channel TEDS.

Irrespective of the time needed to read the transducer data, the NCAP shall wait for at least the duration of the Channel Sampling Period between successive triggers. The Channel Sampling Period is in the Channel TEDS.

The normal behavior of a sensor is illustrated by the state diagram of Figure 4. The concurrent quiescent and triggered states of Figure 3 are shown for reference.

 Copyright © 1998 IEEE. All rights reserved.

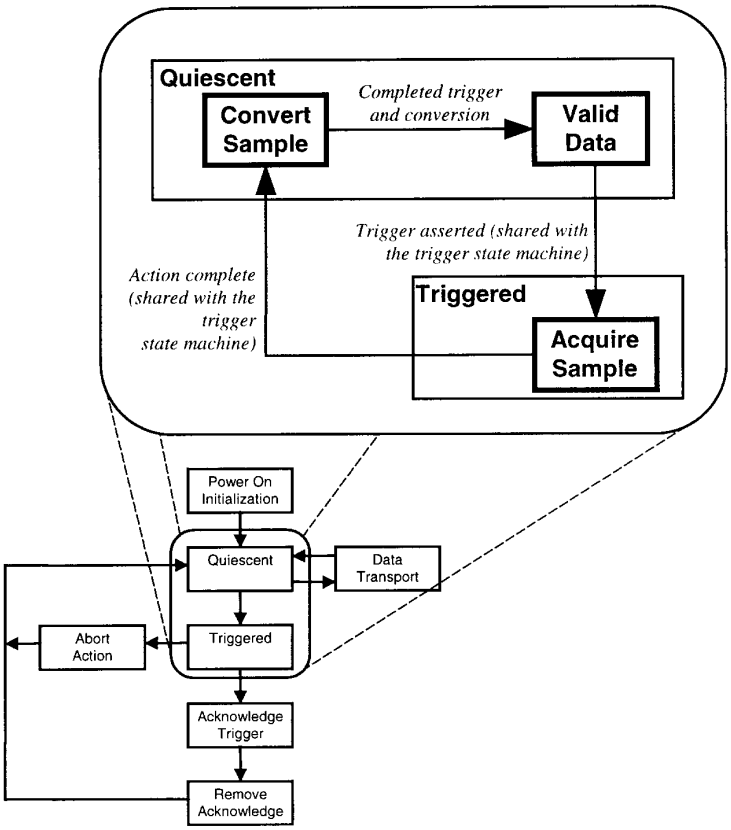

Figure 4—Sensor activity concurrent with the quiescent and triggered states

Table 12 shows the states related to sensors.

Table 12—Sensor states

State	Definition
Acquire Sample	The sample is being acquired. The action is complete when the sample acquisition is complete, irrespective of any further digitization.
Convert Sample	The STIM channel is digitizing the sample and moving it to the transducer data buffer. The STIM leaves this state when all conditions for valid data are met.The Channel Read Setup Time is the time spent in this state.
Valid Data	Valid data from the triggered channel is available to be read.

4.6.5 Triggering actuators

The trigger signal shall cause an actuator channel to assume a new state or to step through a set of states. See 4.6.11 for details on actuators with data sets greater than a single sample. The data set associated with the new state shall have been written to the actuator channel previous to the trigger. Trigger shall not occur until the Channel Write Setup Time has passed following the time the new data is written. The Channel Write Setup Time is in the Channel TEDS.

The STIM shall send a trigger acknowledgment at the time that an actuator channel begins to assume the new state or the first of the set of new states.

Copyright © 1998 IEEE. All rights reserved.

Irrespective of the time necessary to write new data to the actuator, the NCAP shall wait for at least the duration of the Channel Sampling Period between successive triggers. The Channel Sampling Period is in the Channel TEDS.

The normal behavior of an actuator is illustrated by the state diagram in Figure 5. The concurrent quiescent and triggered states of Figure 3 are shown for reference.

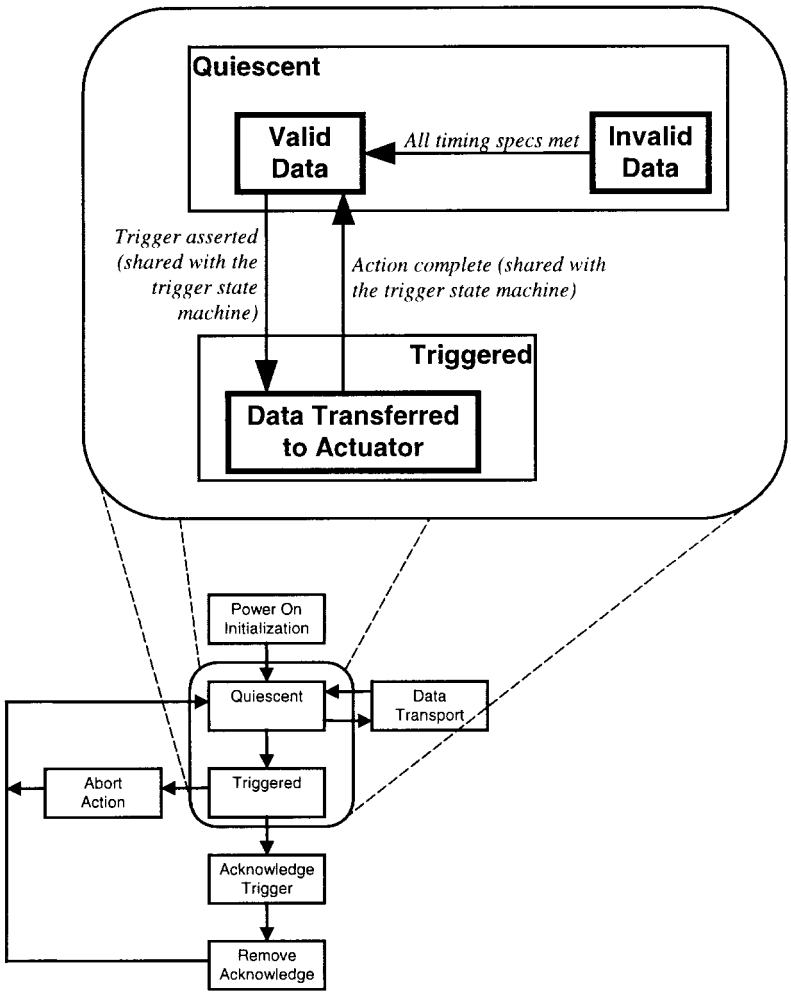

Figure 5—Actuator activity concurrent with the quiescent and triggered states

Table 13 shows the states related to actuators.

Table 13—Actuator states

State	Definition
Data Transferred To Actuator	The last data written to the actuator channel is being transferred to the actuator output buffer.
Invalid Data	The STIM is processing newly received data. The STIM leaves this state when the Channel Write Setup Time and Channel Sampling Period restrictions are met.
Valid Data	Valid data is available to the actuator channel. The channel may be triggered.

Copyright © 1998 IEEE. All rights reserved.

4.6.6 Triggering buffered sensors

The trigger signal shall cause a buffered sensor channel to acquire a new data set, and also to make a previously acquired data set available to be read.

The acquisition of data proceeds concurrently with all other STIM activities. When the acquisition is complete, the data is placed in a holding buffer and further data acquisition waits for the next trigger. Note that the sample time can only be known approximately.

For a buffered sensor channel, the STIM shall send a trigger acknowledgment coincident with the availability of the previously acquired data set. Subsequent to this trigger acknowledge and the additional duration specified by the Channel Read Setup Time, the data shall be available to the NCAP. The Channel Read Setup Time is specified in the Channel TEDS. These times are usually very short.

If another trigger is received before the acquisition of the new sample of data is complete, the acknowledge signal shall not be returned until the data acquisition process is complete. To obtain the fastest trigger cycle time, the NCAP should wait for at least the duration of the Channel Sampling Period between successive triggers. The Channel Sampling Period is in the Channel TEDS.

After initialization or reset, a buffered sensor must be triggered twice before reading the first data. The first trigger samples data and loads the holding buffer and the second makes the data available.

The normal behavior of a buffered sensor is illustrated by the state diagram in Figure 6. The concurrent quiescent and triggered states of Figure 3 are shown for reference. There are two state machines in addition to the trigger state machine. One samples and converts data and the other makes the data available to be read as a result of triggering.

Copyright © 1998 IEEE. All rights reserved.

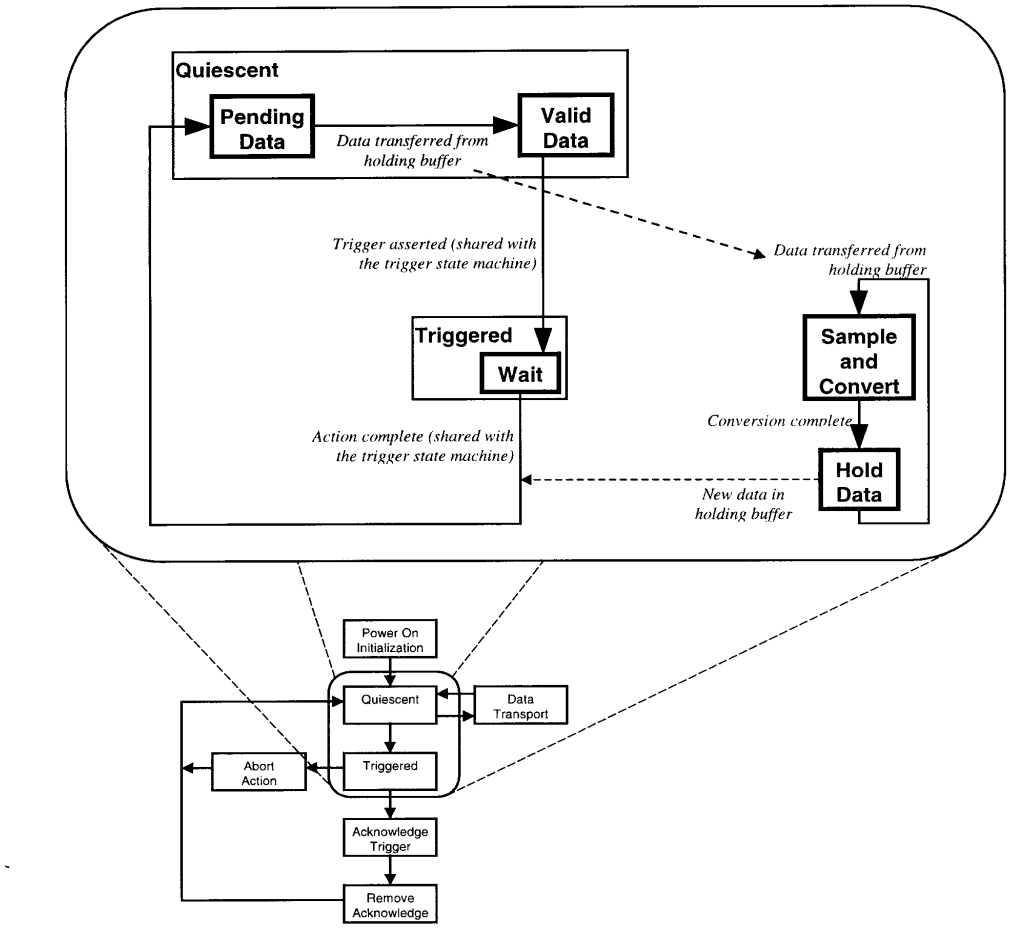

Figure 6—Buffered sensor actions concurrent with the quiescent and triggered states

Table 14 shows the states related to buffered sensors.

Table 14—Buffered sensor states

State	Definition
Sample And Convert	Acquire a new sample and convert to digital form.
Hold Data	Place digitized data into the holding buffer and wait for trigger.
Wait	Wait until the data acquisition process started by the previous trigger is complete. No time is spent in this state if data is already in the holding buffer.
Pending Data	Take the data acquired after the previous trigger and transfer it from the holding buffer to the transducer data buffer to be read. The Channel Read Setup Time is the time spent in this state.
Valid Data	Valid data from the triggered channel is available to be read.

Copyright © 1998 IEEE. All rights reserved.

4.6.7 Triggering data sequence sensors

A data sequence sensor channel shall continuously acquire new data or data sets when enabled, with the timing determined by the STIM. The sample timing may or may not be periodic. The *channel data/event* status bit shall be set at the sample time.

After a trigger is received by a data sequence sensor channel, the STIM shall send a trigger acknowledgment coincident with the next data sample time. Subsequent to this trigger acknowledgment and the additional duration specified by the read setup time, the data associated with this sample time shall be available to the NCAP. The Channel Read Setup Time is specified in the Channel TEDS.

Sampling shall be enabled or disabled by control commands defined in 4.7. A disabled data sequence sensor shall neither respond to a trigger in any way, nor hold off global trigger acknowledgments.

When a STIM contains multiple enabled data sequence channels, it is recommended that all such channels be operated synchronously. This makes global triggering unambiguous. See 4.6.10 for more details about global triggering. The global triggering response of multiple data sequence channels operating asynchronously is beyond the scope of this standard.

In most applications of data sequence sensors, it is desirable to read a number of contiguous single data samples. The *channel missed data or event* status bit shall be set for an enabled data sequence sensor coincident with any data sample time if the trigger is not asserted. The exception is that when the status bit has been cleared, missed samples shall not be detected and the channel missed data or event status bit shall not be set until after the first trigger cycle.

The normal behavior of a data sequence sensor is illustrated by the state diagram in Figure 7. The concurrent quiescent and triggered states of Figure 3 are shown for reference. There are two state machines in addition to the trigger state machine. One continuously samples and converts data and the other makes the data available to be read as a result of triggering.

Copyright © 1998 IEEE. All rights reserved.

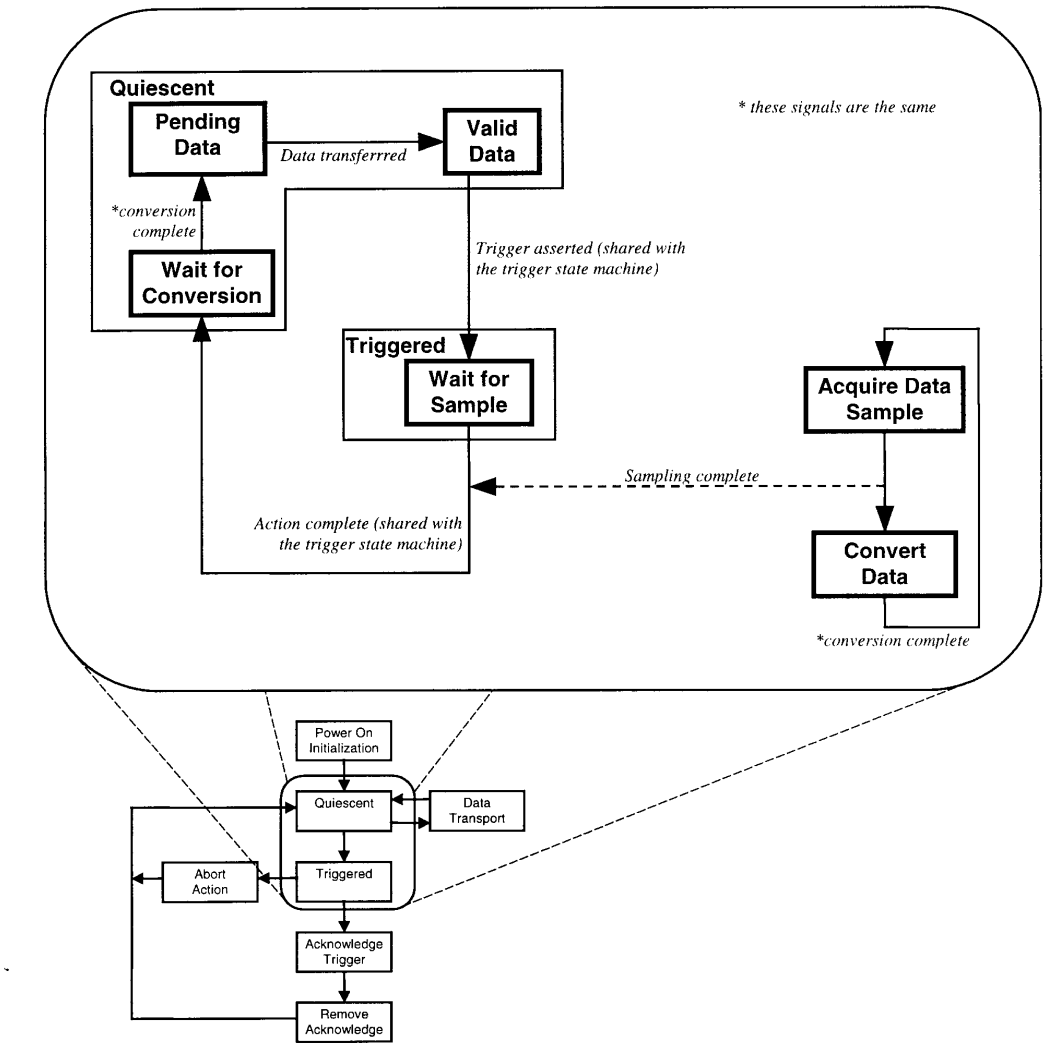

Figure 7—Data sequence sensor actions and the quiescent and triggered states

Table 15 shows the states related to data sequence sensors.

Table 15—Data sequence sensor states

State	Definition
Acquire Data Sample	The sample is being acquired.
Convert Data	The sampled data is digitized.
Wait For Sample	Wait until the next sample is acquired.
Wait For Conversion	Wait in this state until conversion is complete and data is available.
Pending Data	Transfer data to the transducer data buffer to be read. The Channel Read Setup Time is the time spent in this state and in the Wait For Conversion state.
Valid Data	Valid data from the triggered channel is available to be read.

Copyright © 1998 IEEE. All rights reserved.

4.6.8 Triggering buffered data sequence sensors

A buffered data sequence sensor channel shall continuously acquire new data or data sets when enabled, with the timing determined by the STIM. The sample timing may or may not be periodic. When each data acquisition is complete, the STIM shall place the data in a holding buffer, set the *channel data/event* status bit, and begin the next data acquisition. A trigger shall make the data in the holding buffer available to be read.

For a buffered data sequence sensor channel, the STIM shall send a trigger acknowledgment coincident with the availability of the previously acquired data set. Subsequent to this trigger acknowledge and the additional duration specified by the Channel Read Setup Time, the data shall be available to the NCAP. The Channel Read Setup Time is specified in the Channel TEDS. These times are usually very short.

If another trigger is received after the holding buffer is read but before the acquisition of the next sample of data is complete, the acknowledge signal shall not be returned until the data acquisition process is complete.

Sampling shall be enabled or disabled by control commands defined in 4.7. A disabled data sequence sensor shall neither respond to a trigger in any way, nor hold off global trigger acknowledgments.

When a STIM contains multiple enabled buffered data sequence channels, it is recommended that all such channels be operated synchronously. This makes global triggering unambiguous. See 4.6.10 for more details about global triggering. The global triggering response of multiple buffered data sequence channels operating asynchronously is beyond the scope of this standard.

In some applications of buffered data sequence sensors it is desirable to read a number of contiguous single data samples. The *channel missed data or event* status bit shall be set for an enabled buffered data sequence sensor if a data sample overwrites data in the holding buffer that has not been read. The exception is that when the status bit has been cleared, missed samples shall not be detected and the channel missed data or event status bit shall not be set until after the first trigger cycle.

The normal behavior of a buffered data sequence sensor is illustrated by the state diagram in Figure 8. The concurrent quiescent and triggered states of Figure 3 are shown for reference. There are two state machines in addition to the trigger state machine. One continuously samples and converts data and the other makes the data available to be read as a result of triggering

Copyright © 1998 IEEE. All rights reserved.

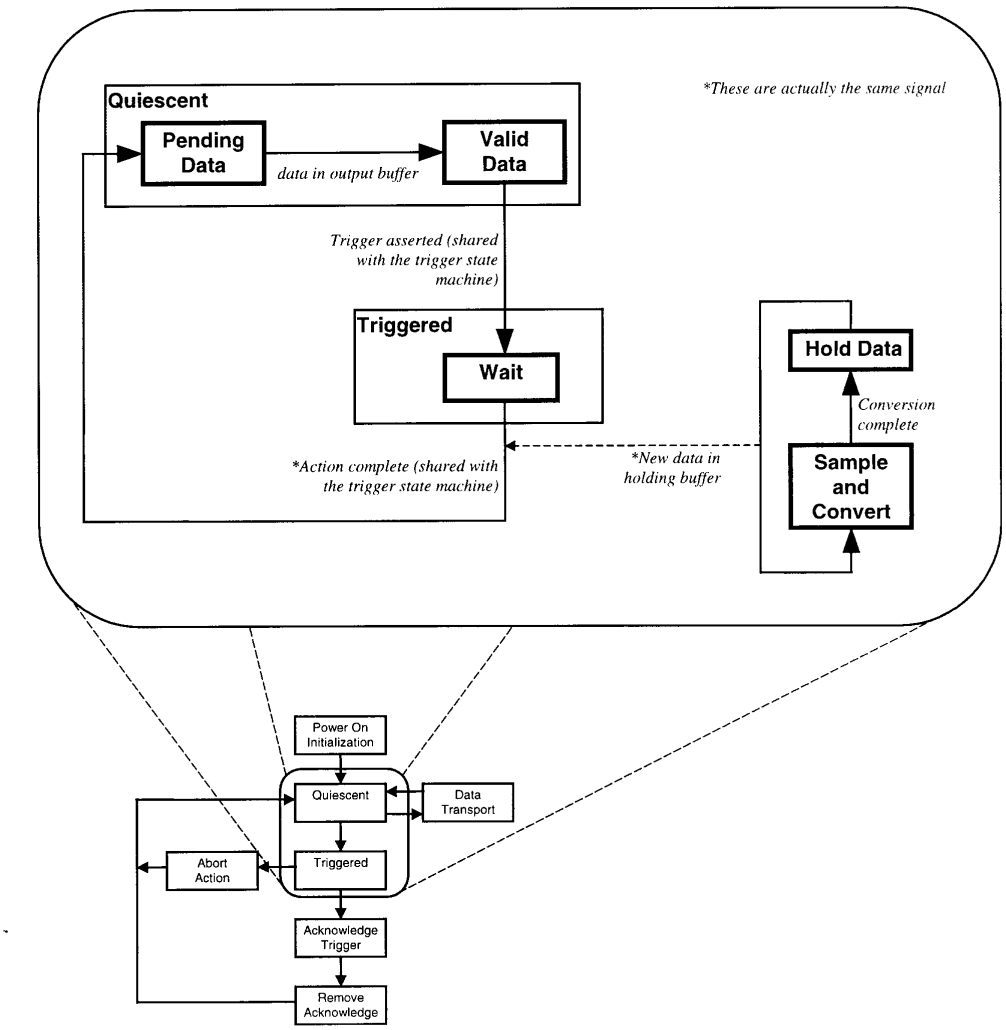

Figure 8—Buffered data sequence sensor states concurrent with the quiescent and triggered states

Table 16 shows the states related to buffered data sequence sensors.

Table 16—Buffered data sequence sensor states

State	Definition
Sample and Convert	Acquire and digitize a new set of data.
Hold Data	Save the digitized data set in the holding buffer. When done, proceed to Sample and Convert state.
Wait	Wait until the data acquisition process started by the previous trigger is complete. No time is spent in this state if data is already in the holding buffer.
Pending Data	Transfer data from the holding buffer to the transducer data output buffer to be read. The Channel Read Setup Time is the time spent in this state.
Valid Data	Valid data from the triggered channel is available to be read.

Copyright © 1998 IEEE. All rights reserved.

4.6.9 Triggering event sequence sensors

An enabled event sequence sensor shall respond to a trigger signal by asserting the trigger acknowledge signal at the occurrence of the next event. A trigger shall have no effect on an event sequence sensor in its *disabled* or *configuration* state, but shall be acknowledged.

An enabled event sequence sensor shall set the *channel data/event* status bit whenever an event is detected irrespective of triggering, and shall additionally set the *channel missed data or event* status bit coincident with any event if the trigger is not asserted. The exception is that the *channel missed data or event* status bit shall not be set until after the first trigger cycle after it is cleared.

An *event* is defined as a transition of state. The allowed states are one or zero. The event sequence sensor may be configured to report rising transitions (from zero to one), falling transitions (from one to zero), or both. The time of the event shall be the point of completion of a selected transition type.

An event sequence sensor may physically consist of one or more digital inputs or an analog input.

a) For a digital event sequence sensor, a rising transition shall be when the inputs match a specific digital pattern without change for a fixed period of time determined by the manufacturer. A falling transition shall be when the inputs cease to match the pattern for the same fixed time.

b) For an analog event sequence sensor, an upper and lower threshold are defined. The lower threshold differs from the upper by a quantity called hysteresis. A rising transition shall be when the analog input value passes successively through the lower and upper thresholds without recrossing either. A falling transition shall be when the analog input value passes successively through the upper and lower thresholds without recrossing either.

An event sequence channel shall have three operational states: *enabled*, *disabled*, and *configuration*. These states shall be set by control commands defined in 4.7. When disabled, an event sequence channel shall operate like a sensor channel and not set the *channel data/event* or *channel missed data or event* status bits. Accessing *read transducer data* from an enabled or disabled event sequence sensor channel shall return a byte with integer value one if the last transition was rising and zero if it was falling.

In the *configuration* state, the event sequence sensor channel shall behave like an actuator, and a single byte written to *write transducer data* allows the NCAP to select which transitions to report as events, according to the enumeration in Table 17. Accessing *read transducer data* shall return the previously written configuration. Triggering shall not be necessary and shall have no effect.

Table 17—Enumeration of event sequence sensor configurations

Value	Meaning
0	Reserved
1	Rising transitions reported
2	Falling transitions reported
3	Both transitions reported
4–255	Reserved

The minimum time between events that can be detected by the channel shall be specified in the Channel TEDS field, Channel Sampling Period (5.2.3.24).

Copyright © 1998 IEEE. All rights reserved.

The normal behavior of an event sequence channel is illustrated by the state diagram in Figure 9. The concurrent triggered state of Figure 3 is shown for reference. The "event happens" signal coincides with "action complete."

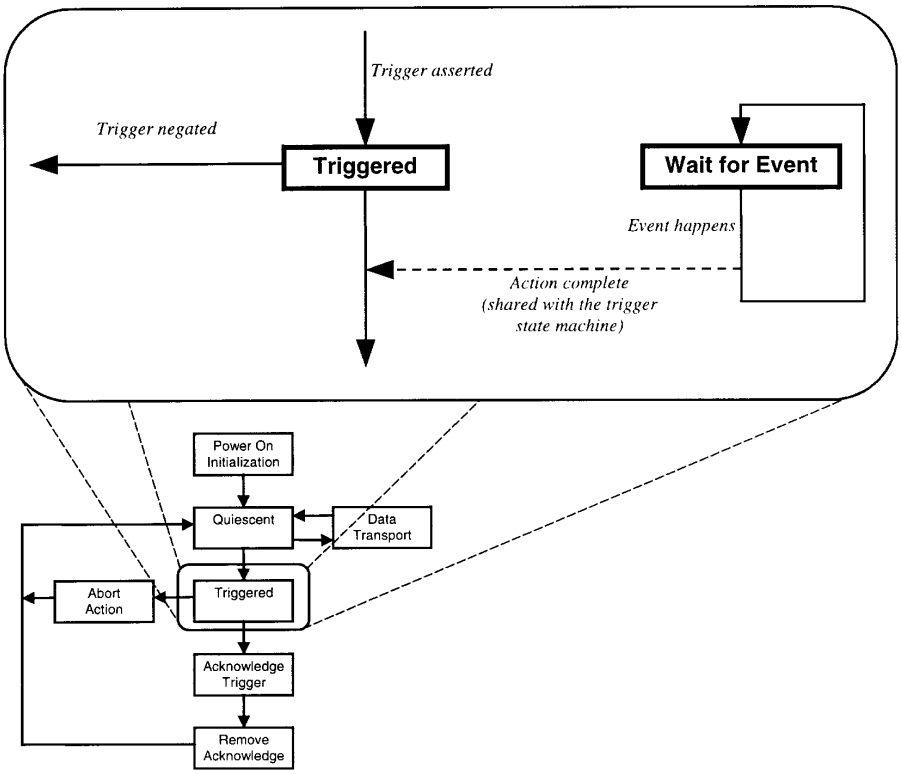

Figure 9—Event sequence sensor states concurrent with the triggered state

Table 18 shows the state associated with event detection.

Table 18—Event detection state machine

State	Definition
Wait for event	The event detection process is waiting for an event to happen.

Extended configuration of an event sequence sensor is possible by means of auxiliary channels. These channels are associated with the event sequence sensor channel by means of groupings reported in the Meta-TEDS (see 4.12 and 5.1.3). The Event Sequence Options field of the Channel TEDS (5.2.3.29) is used to denote which, if any, of these extended configuration options apply.

A digital event sequence sensor may be associated with a sensor that returns the digital input value and/or an actuator that allows a changeable pattern to match. The Channel TEDS for these auxiliary channels shall be appropriate to their function.

An analog event sequence sensor may be associated with a sensor that returns the analog input value and either one or both of two actuators, one to set the upper threshold and one to set the hysteresis. The hysteresis

 Copyright © 1998 IEEE. All rights reserved.

is subtracted from the upper threshold to give the lower threshold. All of these auxiliary channels shall have TEDS appropriate to their function and may have Calibration TEDS associated with them.

Consistency checks may be made by the STIM if the pattern is changeable for a digital event sequence sensor channel or either the threshold or hysteresis is changeable for an analog event sequence sensor channel. Inconsistent values are signaled by the *hardware error status bit*. If inconsistent values are checked, the check shall be made immediately following a change in any of these parameters. The check shall consist of verifying the following relationships:

— For analog values:
maximum sensed input ≥ threshold ≥ (threshold – hysteresis) ≥ minimum sensed input value
— For digital patterns: the pattern is a legal input.

NOTE—Consistency checking by the STIM must perforce only use data in the data model specified by the Channel TEDS of the auxiliary channels.

4.6.10 Global triggering

When the triggered channel address is CHANNEL_ZERO, then all channels shall respond to a trigger signal. Due to the many possible permutations of transducer types, it is not possible to specify the response to global triggering in the general case. Several selected cases are specified below. Behavior of any other permutation is left to the manufacturer.

4.6.10.1 Global triggering with mixed sensors, actuators, and buffered sensors

Trigger acknowledgment shall be held off until all channels have completed their appropriate action. Thus, the trigger acknowledgment is sent by the slowest channel. If global writing of transducer data precedes triggering, or global reading follows, then the worst-case setup times specified by Meta-TEDS must be honored in order to ensure predictable transducer data or actuator states.

The Timing Correction (5.2.3.27) and Trigger Accuracy (5.2.3.28) fields in the Channel TEDS can be used to calculate the timing for all but the slowest channel.

It is recommended that the Timing Correction field be used as follows:

a) For all sensor or actuator channels, the interval between trigger and trigger acknowledgment should be fixed for each channel. The Timing Correction field for each channel should contain the difference between this time interval and that of the slowest channel. Thus, this will be a negative number, or zero (for the slowest channel). The NCAP can then add this number to the global trigger acknowledge time to obtain the time when the channel would have acknowledged trigger.

b) For all synchronously clocked data sequence channels, there may be some skew between sample times. It is recommended that manufacturers should designate one channel to respond to global triggers if this timing feature is used. The entry in the Timing Correction field for this reference channel should be zero, and the entry for all other channels should be their data clock times relative to the reference channel. These entries will all be negative, indicating that these channels are sampled before the reference channel for any particular trigger. Any other channel types included on such a STIM should be sensors, buffered sensors, buffered data sequence sensors, or actuators that respond virtually instantaneously to any trigger, and their Timing Correction fields should be set to NaN.

For event sequence sensors and for sensors or actuators for which time is unimportant, the Timing Correction field should be set to NaN.

Copyright © 1998 IEEE. All rights reserved.

4.6.10.2 Global triggering when enabled event sequence sensors are present

If more than one enabled event sequence channel is present on a STIM that is globally triggered, trigger acknowledgment shall be asserted by the enabled channel first detecting an event. The ambiguity in trigger acknowledgments shall be resolved by means of the *channel data/event* status bit.

If one or more enabled event sequence sensors are present, only the enabled event sequence sensors shall respond to a global trigger. The trigger acknowledgment produced shall also serve as a virtual global trigger to all other non-event sequence channels. No trigger acknowledgment shall be produced by this virtual trigger of other channels. This allows "other" channels to capture state at the time of event detection without requiring NCAP mediation. The actual time of sampling (as opposed to the "triggering") of these non-event sequence channels may be calculated approximately by adding the Worst-Case Channel Update Time (5.1.3.13) to the time of the trigger acknowledgment and using this as the time when the other channels would have generated a trigger acknowledgment.

4.6.11 Triggering channels with Channel Data Repetitions greater than zero

Transducer types sensor, buffered sensor, data sequence sensor, buffered data sequence sensor, and actuator may have Channel Data Repetitions (5.2.3.17) greater than zero. This allows for vectors of data associated with equally spaced increments of some physical parameter that forms an independent variable. The increment, origin, and units associated with this independent variable are listed in the Channel TEDS fields: Series Increment, Series Origin, and Series Units, respectively. Fixed-length time series may be implemented by setting the value of the Series Units field to seconds. The number of instances of data in the com-

 Copyright © 1998 IEEE. All rights reserved.

6.4 Timing

This subclause specifies timing for the data transport and trigger lines.

The data transport rate (the nominal frequency for DCLK) may be varied by the NCAP. All NCAPS and all STIMS shall support a common data rate of 6000 bits/s. The STIM shall support a maximum data rate that is at least this fast; the NCAP shall support a data rate at least this slow. The STIM shall reliably communicate at any data rate slower than its specified maximum. Interruption of DCLK toggling for long periods shall not disrupt STIM communications.

When an NCAP first communicates with a STIM it shall use a data rate (f_{clk}) less than or equal to 6000 bits/s. After the NCAP reads the Meta-TEDS, it may switch to a data rate less than or equal to the maximum data rate specified in the Meta-TEDS.

Timing of the data transport is complicated by the fact that the NCAP is allowed to change the data rate. In order to ensure reliable data transport at the selected data rate, most data-transport-related timing parameters, such as setup and hold times, are based on one of the following parameters:

a) f_{max}: the maximum data rate supported by the STIM, specified in the Meta-TEDS (5.1.3.23).
b) f_{clk}: the data rate selected by the NCAP. DCLK toggles at this rate.
c) f_{clk_max}: the maximum data rate the NCAP is capable of supporting. This is known to the NCAP software in a manner beyond the scope of this standard.
d) f_{clk_min}: the minimum data rate the NCAP is capable of supporting. This is known to the NCAP software in a manner beyond the scope of this standard.

Table 53 lists timing requirements for the NCAP. See Figure 17 for definitions of NCAP timing parameters.

Table 53—NCAP timing requirements

Characteristic	Symbol	Unit	Requirement
DCLK rate	f_{clk}	bits per second	$f_{clk} \le f_{max}$
DCLK high time	t_{ch}	seconds	$t_{ch} \ge \frac{0.2}{f_{clk}}$
DCLK low time	t_{cl}	seconds	$t_{cl} \ge \frac{0.45}{f_{clk}}$
DOUT valid to DCLK setup time	t_{sun}	seconds	$t_{sun} \le \frac{0.2}{f_{clk\_max}}$
DCLK to DOUT invalid hold time	t_{hn}	seconds	$t_{hn} \le \frac{0.2}{f_{clk\_max}}$
DCLK to DIN valid delay	t_{dn}	seconds	$t_{dn} \le \frac{0.2}{f_{clk\_max}}$
Rise or fall time: DCLK	t_{edgex}	seconds	$t_{edgex} \le \frac{0.05}{f_{clk\_max}}$
Rise or fall time: DCLK, DIN, NIOE, NTRIG	t_{edgen}	microseconds	$t_{edgen} \le 2$

Copyright © 1998 IEEE. All rights reserved.

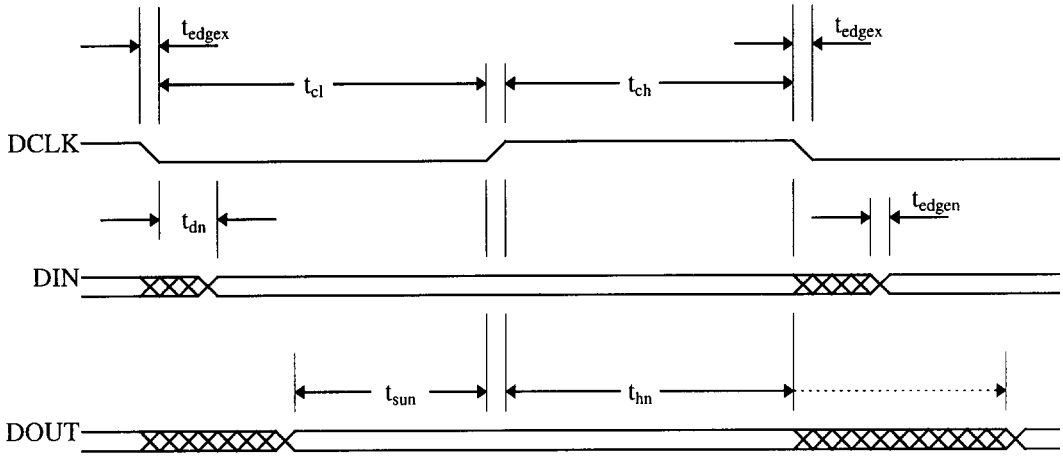

Figure 17—NCAP timing

The first three requirements in Table 53 apply to operational systems in which the NCAP is changing the data rate, and thus either directly or indirectly changing DCLK high and low times. The minimum DCLK high and low times given are sufficient to ensure that all setup and hold times are met when a STIM is directly connected to an NCAP, provided that all other requirements in Tables 53 and 54 are met by the STIM and NCAP individually. Within the timing restrictions below, DCLK need not have a constant frequency or duty cycle.

The NCAP manufacturer shall determine t_{sun}, t_{hn}, t_{dn}, t_{edgen}, and t_{edgex} by measuring the NCAP at the TII connection point. *Setup and hold times thus include delays from filtering or protective circuits on NCAP inputs and outputs*. DCLK, DIN, NIOE, and NTRIG shall be loaded by 200 pF capacitance. DOUT shall be supplied by a generator capable of producing rise and fall times less than $0.01/f_{clk\_max}$ seconds. The NCAP manufacturer shall choose f_{clk_max} so that all the requirements of Table 53 are met.

Table 54 lists timing requirements for the STIM. See Figure 18 for definitions of STIM timing parameters.

Table 54—STIM timing requirements

Characteristic	Symbol	Unit	Requirement
Maximum Data Rate	f_{max}	bits per second	$f_{max} \geq 6000$
DIN valid to DCLK setup time	t_{su}	seconds	$t_{su} \leq \frac{0.2}{f_{max}}$
DCLK to DIN invalid hold time	t_h	seconds	$t_h \leq \frac{0.2}{f_{max}}$
DCLK to DOUT valid delay	t_d	seconds	$t_d \leq \frac{0.2}{f_{max}}$
Rise or fall time: DOUT, NACK, NINT	t_{edges}	microseconds	$t_{edges} \leq 2$

Copyright © 1998 IEEE. All rights reserved.

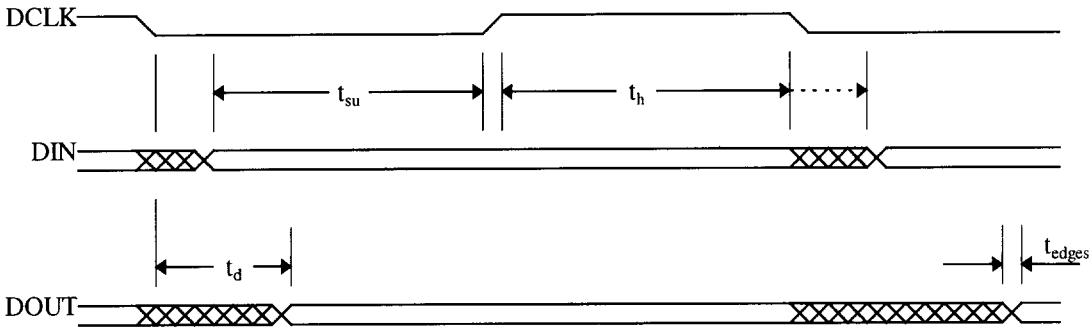

Figure 18—STIM timing

The STIM manufacturer shall determine t_{su}, t_h, t_d, and t_{edges} by measuring the STIM at the TII connection point. *Setup and hold times thus include delays from filtering or protective circuits on STIM inputs and outputs*. DOUT, NACK, and NINT shall be loaded by 200 pF capacitance. DCLK and DIN shall be supplied by a generator capable of producing rise and fall times less than $0.01/f_{max}$ seconds. The STIM manufacturer shall choose f_{max} so that all the requirements of Table 54 are met.

NOTE—The requirements of Table 53 and 54 leave a budget of $0.05/f_{clk}$ seconds for timing uncertainty and cable delays. Practically, this limits a system without cabling and with timing uncertainty of 5 ns to a maximum system data rate of 10 Mb/s. A system with cable with round-trip delay of 50 ns is practically limited to a maximum system data rate of 1 Mb/s.

The remaining timing diagrams in this subclause relate to timing parameters found in the Meta-TEDS and Channels TEDS.

STIM Handshake Timing (5.1.3.19) applies to negation of NACK for both data transport and triggering, as illustrated in Figures 19 and 20. NACK may be negated before NIOE is negated, but NACK shall be negated no later than t_{hs} after NIOE is negated.

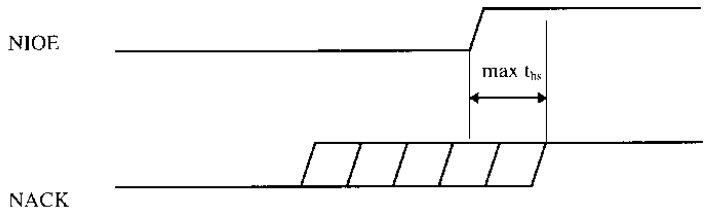

Figure 19—End-of-frame timing

Copyright © 1998 IEEE. All rights reserved.

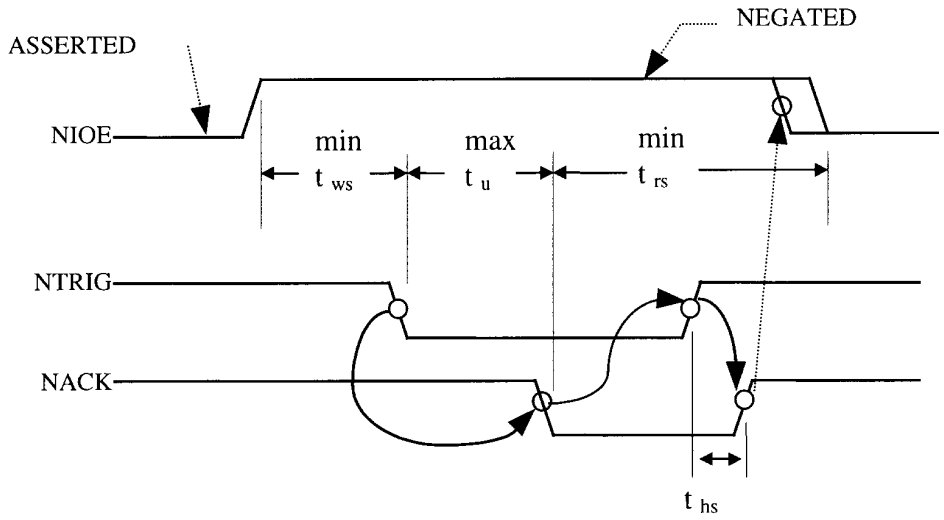

Figure 20—Triggering timing

Figure 20 also defines the timing relationships between data transport involving the triggered channel and triggering of that channel. For clarity, any other data transport is not shown.

a) Minimum t_{ws} shall be the Channel Write Setup Time specified in the Channel TEDS (5.2.3.22), if data has been written to an actuator channel, or zero otherwise. If triggering is on CHANNEL_ZERO, then t_{ws} shall be replaced by t_{gws}, the Global Write Setup Time specified in the Meta-TEDS (5.1.3.14).

b) Minimum t_{rs} shall be the Channel Read Setup Time specified in the Channel TEDS (5.2.3.23) when a sensor channel has been triggered. If no read from the channel is to be executed following the trigger, then NIOE may be asserted as soon as NACK is negated. If triggering is on CHANNEL_ZERO, then t_{rs} shall be replaced by t_{grs}, the Global Read Setup Time specified in the Meta-TEDS (5.1.3.15).

c) Maximum t_u shall be the Channel Update Time specified in the Channel TEDS (5.2.3.21) when a channel has been triggered. If triggering is on CHANNEL_ZERO, then t_u shall be replaced by t_{wu}, the Worst-case Channel Update Time specified in the Meta-TEDS (5.1.3.13).

Figure 21 defines timing restrictions when triggering a previously triggered channel. This timing shall apply to any individual channel regardless of any data transport activity or triggering of any other channel.

a) Minimum t_{sp} shall be the Channel Sampling Period specified in the Channel TEDS (5.2.3.24). If triggering is on CHANNEL_ZERO, then t_{sp} shall be replaced by t_{wsp}, the Worst-Case Channel Sampling Period specified in the Meta-TEDS (5.1.3.16).

b) Irrespective of any other considerations, t_{sp} or t_{wsp} is constrained by the handshake setup times as follows:

$$t_{sp} \geq t_u + t_{hs} \quad \text{or} \quad t_{wsp} \geq t_{wu} + t_{hs}$$

Copyright © 1998 IEEE. All rights reserved.

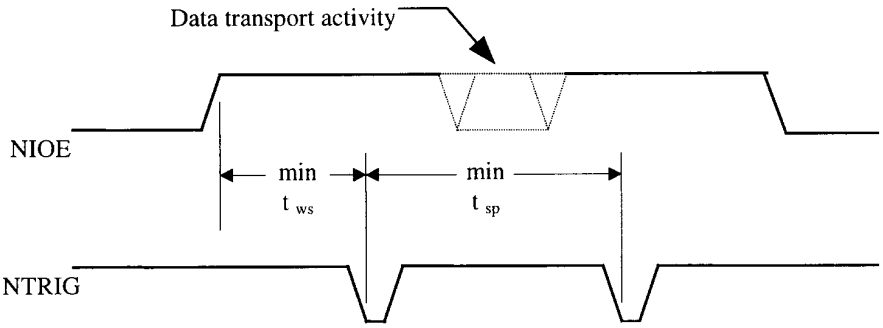

Figure 21—Multiple NTRIG timing

Figure 22 defines the application of the End-Of-Frame Detection Latency (t_{lat}) in the Meta-TEDS (5.1.3.20). By negating NIOE a minimum of t_{lat} seconds between data transport frames the NCAP can ensure that the STIM can detect the end of one frame and the beginning of another.

Figure 22—Data transport latency

The byte hold-off time, t_{hold}, shown in Figure 23, is the time between the last DCLK rising edge of a byte transfer and the NACK transition by the STIM during data transport. The maximum t_{hold} shall be the TEDS Hold-Off Time (5.1.3.21) if the data transport is addressed to TEDS functions, or the Operational Hold-Off Time (5.1.3.22) if the data transport is addressed to operational functions. If the data transport is addressing channel transducer data, the sum of all hold-offs for the entire data transport frame shall not exceed the Channel Aggregated Hold-Off Time (5.2.3.26) in the Channel TEDS. If CHANNEL_ZERO transducer data is addressed, the sum of all hold-offs shall not exceed the sum of all Channel Aggregated Hold-Off Times. If any of these hold-off time restrictions are violated, the NCAP may assume that the addressed function is not working properly and may abort the data transport frame.

Copyright © 1998 IEEE. All rights reserved.

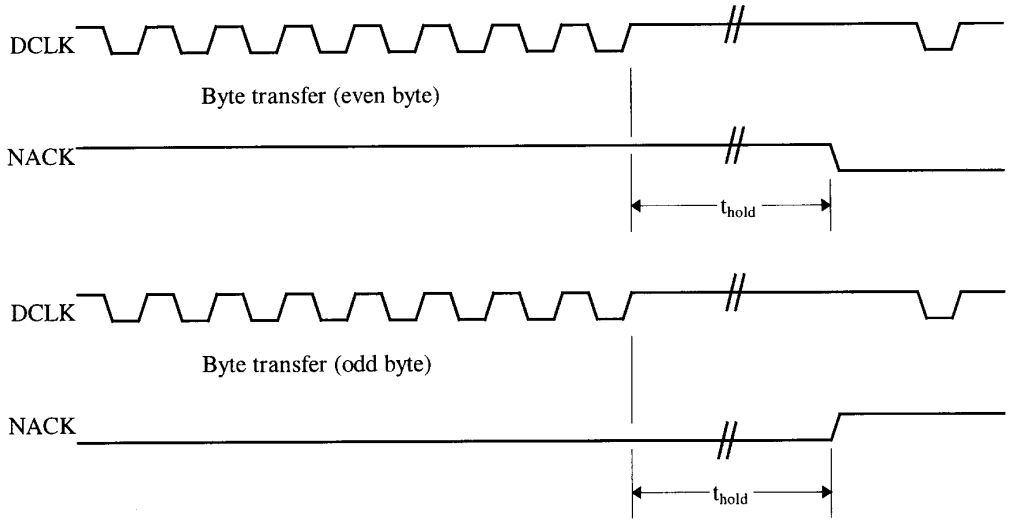

Figure 23—Byte hold-off timing

6.5 Electrical specifications

6.5.1 Power

a) The voltage on the POWER line at the NCAP shall be + 5 V dc ± 0.20 V dc with respect to COMMON.

b) The manufacturer of each STIM shall specify the power it consumes. The STIM shall expect no more than 75 mA at the specified voltage from the NCAP.

c) The STIM may use a separate power supply as necessary; however, power for the STIM interface control circuitry (typically consisting of a microprocessor, FPGA, or ASIC) shall be provided only through the primary communications interface (the TII).

d) In all cases, the signal, common, and power lines of the TII shall be isolated from ground (frame or earth).

e) A STIM shall not be capable of actively sourcing current over the POWER line.

f) It is highly recommended, but not required, that the NCAP should actively control the application of POWER to the STIM. In this case, the NCAP should only apply POWER (typically through a FET switch) only when the STIM is present.

Figure 24 illustrates the active control of the application of POWER to the STIM by the NCAP. In this case, POWER is provided by the NCAP only if the STIM is present (eliminating the possibility of shorts). Note that typically there may be some short delay until insertion or extraction events are recognized by the NCAP.

The following events are shown in Figure 24:

A) An insertion event occurs.
B) The NCAP turns on POWER to the STIM.
C) An extraction event occurs.
D) The NCAP turns off POWER to the STIM.

Copyright © 1998 IEEE. All rights reserved.

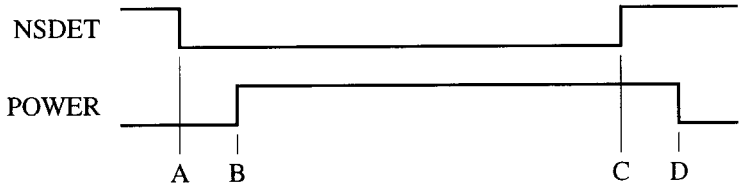

Figure 24—Active control of STIM power

Figure 25 illustrates the non-active control of the application of POWER to the STIM by the NCAP. In this case, POWER is immediately available to the STIM at the same time NSDET is asserted.

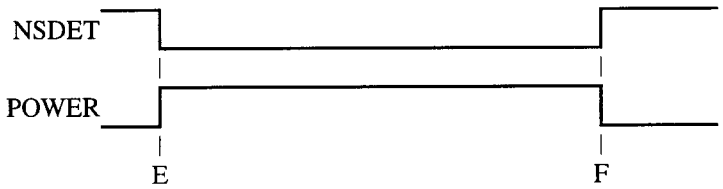

Figure 25—Non-active control of STIM power

The following events are shown in Figure 25:

E) An insertion event occurs and POWER is available to the STIM.
F) An extraction event occurs and POWER is unavailable to the STIM.

Figure 26 illustrates the configuration in which the NCAP supplies power to the STIM interface control circuitry and any measurement and/or actuator circuitry. The STIM need not be concerned where the NCAP gets power.

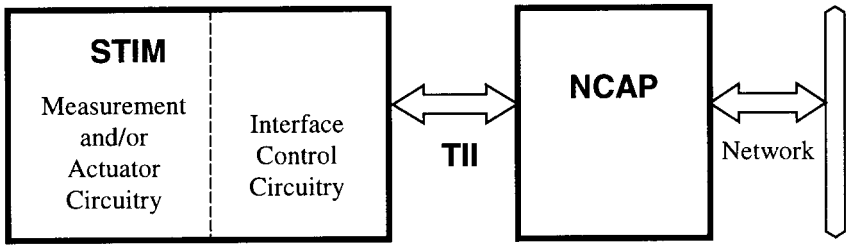

Figure 26—NCAP supplies power to the STIM interface control circuitry and the STIM measurement and/or actuator circuitry

Figure 27 illustrates the configuration in which the NCAP supplies power to the STIM interface control circuitry, but the power needed by the measurement and/or actuator circuitry exceeds the current and/or voltage levels that can be supplied by the primary communications interface. In this case, power for the measurement and/or actuator circuitry can be supplied from a secondary power connector. The STIM need not be concerned about the source of optional secondary power.

Copyright © 1998 IEEE. All rights reserved.

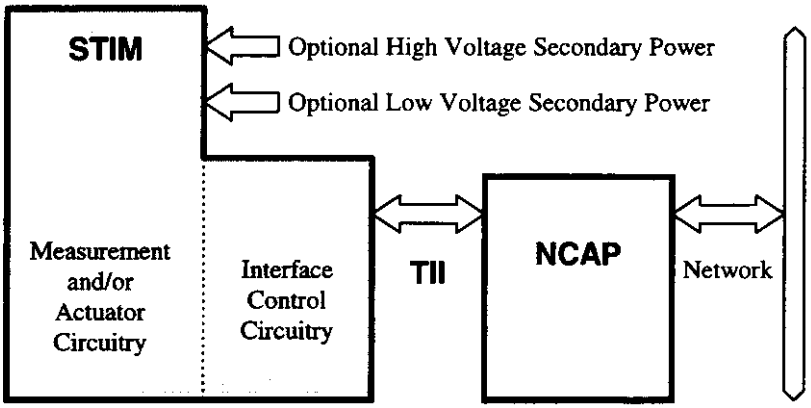

Figure 27—NCAP supplies power to the STIM interface circuitry; the optional secondary connector supplies power to the STIM measurement and and/or actuator circuitry.

6.5.2 Logic signals

This subclause specifies the logic levels, drive, and load characteristics that shall apply to the logic signals designated DOUT, DIN, DCLK, NIOE, NTRIG, NACK, and NINT. These specifications are based on nominal 5 V dc CMOS logic.

6.5.2.1 Logic levels

At any signal input on either the NCAP or the STIM, a voltage greater than 0.8 times the POWER node voltage shall be interpreted as logic HIGH, and a voltage less than 0.15 times the POWER node voltage shall be interpreted as logic LOW.

NOTE—The common CMOS logic thresholds (70% and 30% of the power supply voltage) and the common TTL logic thresholds (2 V dc and 0.8 V dc) already comply with these requirements.

6.5.2.2 Drive and loading

a) Input loads and output drive are defined here at specific voltages to ensure compatibility. Outputs are assumed to be within the voltage range defined by the POWER and COMMON nodes. This standard does not define the permissible voltages outside this range for inputs.

b) Any output driving towards the logic HIGH state shall be capable of sourcing a minimum of 200 μA at output voltages less than 0.9 times the POWER node voltage.

c) Any input shall sink no more than 20 μA at voltages greater than 0.9 times the POWER node voltage.

d) Any output driving towards the logic LOW state shall be capable of sinking a minimum of 1.5 mA at output voltages greater than 0.1 times the POWER node voltage.

e) Any input shall source no more than 600 μA at voltages less than 0.1 times the POWER node voltage.

f) NCAP and STIM outputs shall meet the timing specifications when driving a capacitance of 200 pF.

g) NCAP and STIM inputs shall present a maximum of 100 pF of capacitance.

h) Any cabling between NCAP and STIM shall not exceed 100 pF capacitance between any two wires or between any wire and shield (if a shield is used).

Copyright © 1998 IEEE. All rights reserved.

6.5.2.3 Quiescent states

Signal input lines on the NCAP—namely NINT, NACK, DOUT, and NSDET—shall be passively held high via a large resistance (typically greater than 10 000 Ω) to POWER, so that these signals assume a known logic level if no STIM is attached to the NCAP. The current due to this resistance must be accounted for in the input current sourcing specification given in 6.5.2.2.

6.5.3 Hot-swap capability

Detection of insertion events (when a STIM is attached to a powered NCAP) and extraction events (when a STIM is removed from a powered NCAP) are signaled using the NSDET signal. At the STIM, NSDET shall be connected to COMMON. At the NCAP, NSDET shall be passively pulled up toward POWER and NSDET shall be made available to the NCAP processor. Signal states for NSDET are shown in Table 55.

Table 55—NSDET signal states

Event	NSDET	Action
Insertion	Becomes asserted	Pulled low
Extraction	Becomes negated	Pulled high

All IEEE 1451.2 signal lines driven by the STIM, except NSDET, shall be negated during STIM initialization. An insertion event is detected by the NCAP whenever NSDET is asserted, and an extraction event is detected whenever NSDET is negated. During the mechanical connection of STIMs and NCAPs, the signal on NSDET may oscillate, and may require some time to stabilize.

After POWER is applied to the STIM, the STIM needs some period of time to initialize itself. STIM initialization in this context refers to the STIM getting to the point where it can communicate with the NCAP—warming up transducers is a different kind of delay. The NCAP may assert NIOE any time after NSDET has stabilized in the asserted state and NACK has stabilized in the negated state. Since the NCAP may not continue the transaction until NACK is asserted, the NCAP shall wait until STIM initialization has been completed (since all IEEE 1451.2 signal lines driven by the STIM are negated during STIM initialization) and the STIM is able to respond to NIOE with NACK. The NCAP may then proceed to communicate with the STIM (at the minimum data rate supported by all STIMs). At the end of the transaction, the NCAP negates NIOE and waits until the STIM negates NACK in response.

Figure 28 illustrates the mechanism by which the STIM communicates to the NCAP that STIM initialization has been completed. The following events occur during STIM initialization:

A) An insertion event occurs.
B) NCAP initiates a transaction by asserting NIOE.
C) STIM initialization is complete.
D) STIM asserts NACK in response to NIOE.
E) NCAP negates NIOE to signal the end of the transaction.
F) STIM negates NACK in response to NIOE.

 Copyright © 1998 IEEE. All rights reserved.

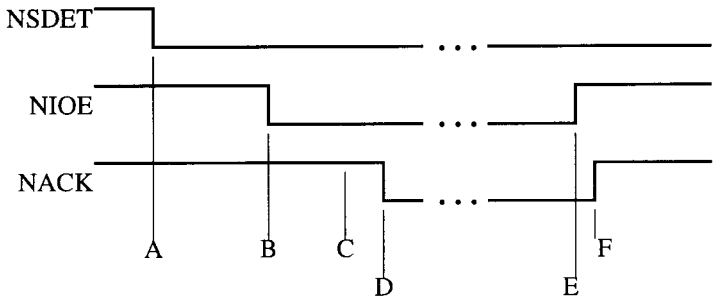

Figure 28—Initial STIM communication timing

6.6 Physical specification

6.6.1 Connectors

The connector pin numbering shown in Table 56 is recommended. Table 56 also shows a recommended wire color-coding for unterminated STIMs.

6.6.1.1 Primary communications connector

It is recommended that the STIM connector be

a) At least ten pins
b) Male (because the STIM will not supply power)
c) Polarized

Table 56—Recommended primary communications connector pin specification

Pin no.	Signal name	Wire color	Direction for NCAP	Direction for STIM
1	DCLK	Brown	OUT	IN
2	DIN	Red	OUT	IN
3	DOUT	Orange	IN	OUT
4	NACK	Yellow	IN	OUT
5	COMMON (GROUND)	Green	POWER	POWER
6	NIOE	Blue	OUT	IN
7	NINT	Violet	IN	OUT
8	NTRIG	Gray	OUT	IN
9	POWER (+5 V dc)	White	POWER	POWER
10	NSDET	Black	IN	OUT

6.6.1.2 Optional secondary power connector

It is recommended that an optional secondary STIM connector be

a) Male (because the STIM will not supply power)
b) Polarized

Copyright © 1998 IEEE. All rights reserved.

Annex A

(informative)

Bibliography

[B1] Hamilton, Bruce, “A Compact Representation of Physical Units” Hewlett-Packard Company, Palo Alto, Calif., Hewlett-Packard Laboratories Technical Report HPL-96-61, 1995.[12]

[B2] NIST Technical Note 1297, 1994 Edition: “Guidelines for Evaluating and Expressing the Uncertainty of NIST Measurement Results” by Barry N. Taylor and Chris E. Kuyatt. Produced by the Physics Laboratory, National Institute of Standards and Technology, United States Department of Commerce, Gaithersburg, Maryland 20899-0001, United States of America. Available from the Superintendent of Documents, United States Government Printing Office, Washington, DC 20402, United States of America.

[12]Available from Hewlett-Packard Laboratories Technical Publications Department, 1501 Page Mill Road, Mail Stop 2L, Palo Alto, California, 94304, (415) 857-4573.

Copyright © 1998 IEEE. All rights reserved.