INTERNATIONAL STANDARD

ISO/IEC 13213

ANSI/IEEE Std 1212

First edition
1994-10-05

Information technology – Microprocessor systems – Control and Status Registers (CSR) Architecture for microcomputer buses

Technologies de l'information – Systèmes à microprocesseurs – Architecture des registres de commande et d'état pour bus de micro-ordinateur

Reference number
ISO/IEC 13213: 1994(E)
ANSI/IEEE
Std 1212, 1994 Edition

Abstract: The document structure and notation are described, and the objectives and scope of the CSR Architecture are outlined. Transition set requirements, node addressing, node architectures, unit architectures, and CSR definitions are set forth. The ROM specification and bus standard requirements are covered.

Keywords: CSR Architecture, bus architecture, bus standard, interoperability, microprocessors, node addressing, registers, transaction sets

---

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

ISBN 1-55937-448-9

October 5, 1994 *SH94220*

ISO/IEC 13213 : 1994
[ANSI/IEEE Std 1212, 1994 Edition]
(Incorporates ANSI/IEEE Std 1212-1991)

Information technology—Microprocessor systems—Control and Status Registers (CSR) Architecture for microcomputer buses

Sponsor

Microprocessor and Microcomputer Standards Committee of the IEEE Computer Society

Adopted as an International Standard by the
International Organization for Standardization
and by the
International Electrotechnical Commission

Published by
The Institute of Electrical and Electronics Engineers, Inc.

American National Standard

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are member of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and nongovernmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC1 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75% of the national bodies casting a vote.

In 1994, ANSI/IEEE Std 1212-1991 was adopted by ISO/IEC JTC1, as draft International Standard ISO/IEC DIS 13213. This edition incorporates editorial comments received in the review of ISO/IEC DIS 13213.

International Organization for Standardization/International Electrotechnical Commission
Case postale 56 • CH-1211 Genève 20 • Switzerland

IEEE Standards documents are developed within the Technical Committees of 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 technical 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

IEEE Standards documents may involve the use of patented technology. Their approval by the Institute of Electrical and Electronics Engineers does not mean that using such technology for the purpose of conforming to such standards is authorized by the patent owner. It is the obligation of the user of such technology to obtain all necessary permissions.

Introduction

(This introduction is not a part of this International Standard or of ANSI/IEEE Std 1212, 1994 Edition.)

Bus standards have often been set by hardware designers who have focused on the transport mechanisms for sending read and write transactions on a bus. Additional software considerations are needed to ensure interoperability between boards, as users of current bus "standards" have discovered. Therefore, many bus standards have been supplemented with one or several *de facto* or recommended register architectures, which have usually differed for each bus standard.

Through the cooperative efforts of the P1394 Serial Bus, P896 Futurebus+, and P1596 Scalable Coherent Interface (SCI) Working Groups, the need for a more formal approach to defining a common scalable bus-technology-independent Control and Status Register (CSR) Architecture was recognized. The hope is that, by sharing a uniform CSR Architecture, these systems will minimize the software and firmware changes when migrating a processor from one system bus to another or when bridging from one bus to another, and that software costs for migrating between standards (as technology evolves) will be reduced. The P1212 CSR Architecture Working Group was fortunate to have the wide range of bus technologies (from approximately 40 Mb/s for Serial Bus to approximately 1 Gbyte/s for SCI) to test the performance and cost scalability of its designs. The popularity of the Futurebus+ standard ensured that the CSR Architecture specification would be reviewed by a large audience for use in a wide variety of applications.

The scope of the CSR Architecture includes the definition of the generic registers needed to initialize, configure, and test nodes within a system. Other broadcast registers are sufficiently standardized to ensure interoperability between modules supplied by different vendors. The CSR document also defines address-space maps, bus transaction sets (reads, writes, and locks), and ROM data formats.

Protocols are defined for interrupting processors, passing messages, and for accurately synchronizing distributed clocks. These definitions are intended to provide a sufficient and standard framework for the design of vendor-dependent unit architectures. Parts of this CSR Architecture are likely to indirectly influence the processor designs of the future.

The following is a list of participants in the IEEE Control and Status Register (CSR) Working Group at the time ANSI/IEEE Std 1212-1991 was approved:

David V. James, *Chair*

Barbara Aichinger
Knut Alnes
Robert S. Baxter
Harrison Beasley
David Brash
Mark Bunker
Robert C. Carpenter
D. Del Corso
Jon Crowell
Stephen Deiss
Ian Dobson
Emer Dooley
Michael A. Dorsett
Sam Duncan
W.P. Evertz
Frank Fidducia
John R. Fortier
Ralph Frangioso
Tony Grigg
David B. Gustavson
Claes-Gôran Gustavsson
Mark Hassel
David Hawley
Hubert Holierhoek
Ed Jacques
Marit Jenssen
Ernst Kristiansen
Ralph Lachenmaier
Jim Leahy
Jim McGrath
Thanos Mentzelopoulous
Jim Moidel
Klaus Mueller
George Nacht
Mitsunori Nakata
Richard Napolitano
Daniel C. O'Connor
Mira Pauker
Chet Pawlowski
James M. Pexa
Mike Roby
Tim Scott
Don Senzig
Patricia Smith
Joanne Spiller
Bob Squirrell
Haruhisa Suzuki
Michael Teener
John Theus
Yoshiaki Wakimura
Mike Wenzel
Martin Whittaker
Hans Wiggers
Dwight Wilcox
Mark Williams
David L. Wright

The following persons were on the balloting committee that approved this document for submission to the IEEE Standards Board:

John Allen
Knut Alnes
Harrison A. Beasley
Kyle M. Black
John Black
Kim Clohessy
Jonathan C. Crowell
Stephen L. Diamond
Samuel Duncan
Wayne Fischer
Joseph D. George
Andy Glew
David B. Gustavson
Zoltan R. Hunor
John Hyde
Ed Jacques
David V. James
Kenneth Jansen
Hubert Kirrmann
Ernst H. Kristiansen
Gerry Laws
James D. Mooney
Klaus Dieter Mueller
Gary A. Nelson
J. D. Nicoud
Daniel C. O'Connor
Mira Pauker
Donald Pavlovich
Richard Rawson
Carl Schmiedekamp
Don Senzig
Paul Sweazey
Michael G. Thompson
Eike Waltz
Hans A. Wiggers
Mark Williams
Oren Yuen

When the IEEE Standards Board approved this standard on December 5, 1991, it had the following membership:

Marco W. Migliaro, *Chair*
Donald C. Loughry, *Vice Chair*
Andrew G. Salem, *Secretary*

Dennis Bodson
Paul L. Borrill
Clyde Camp
James M. Daly
Donald C. Fleckenstein
Jay Forster*
David F. Franklin
Ingrid Fromm
Thomas L. Hannan
Donald N. Heirman
Kenneth D. Hendrix
John W. Horch
Ben C. Johnson
Ivor N. Knight
Joseph L. Koepfinger*
Irving Kolodny
Michael A. Lawler
John E. May, Jr.
Lawrence V. McCall
Donald T. Michael*
Stig L. Nilsson
John L. Rankine
Ronald H. Reimer
Gary S. Robinson
Terrance R. Whittemore

*Member Emeritus

Also included are the following nonvoting IEEE Standards Board liaisons:

Fernando Aldana
Satish K. Aggarwal
James Beall
Richard B. Engelman
Stanley Warshaw

IEEE Std 1212-1991 was approved by the American National Standards Institute on May 14, 1992.

This page intentionally left blank

Contents

CLAUSE PAGE

CLAUSE PAGE

ANNEXES

ANNEXES PAGE

Information technology—Microprocessor systems—Control and Status Registers (CSR) Architecture for microcomputer buses

1. Document structure and notation

1.1 Document structure

This International Standard defines the address-space maps, the bus transaction sets, and the node's CSRs. The intention is to provide a sufficient and standard framework for the design of vendor-dependent unit architectures.

The specification includes the format and content of the configuration ROM on the node. The configuration ROM provide the parameters necessary to autoconfigure systems with nonprocessor nodes provided by multiple vendors.

Note that a monarch selection process, which selects one processor to boot the system, is not defined. A monarch selection process would be necessary to initialize a system containing processors provided by different vendors.

The annexes provide background for understanding the usage of this CSR Architecture specification. The CSR Architecture provides the specification upon which conforming designs should be based. The annex clauses illustrate ways that these capabilities could be used. Note that the annexes are nonbinding.

1.2 References

The following standards contain provisions which, through reference in this text, constitute provisions of this International Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ANSI/ISO/IEC 9899:1990, Programming Languages—C.[1,2]

ISO/IEC 646:1991, Information technology—ISO 7-bit coded character set for information interchange.[2]

1.3 Conformance levels

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

1.3.1 expected: A key word used to describe the behavior of the hardware or software in the design models assumed by the CSR Architecture. Other hardware and software design models can also be implemented.

1.3.2 may: A key word that indicates flexibility of choice with no implied preference.

---

[1] Replaces ANSI X3.159-1989.

[2] ISO documents are available from ISO Central Secretariat, 1 rue de Varembé, Case Postale 56, CH-1211, Genève 20, Switzerland/Suisse; and from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036-8002, USA.

1.3.3 shall: A key word indicating a mandatory requirement. Designers are required to implement all such mandatory requirements to ensure interoperability with other CSR Architecture conformant products.

1.3.4 should: A key word indicating flexibility of choice with a strongly preferred alternative. The phrase *it is recommended* has the same meaning.

1.4 Technical glossary

A large number of bus and interconnect-related technical terms are used in the CSR Architecture document. These terms are described herein:

1.4.1 active test: An ongoing test that is invoked by a write to the TEST_START register. The node is in the testing state (STATE_CLEAR.*state* is equal to testing) while an active test is in progress.

1.4.2 address_error: An error-status code returned to the requester when a transaction is directed to a non-existing address; on some buses, this has been called a NACK (negative acknowledge). The address_error status is generally returned if a valid address acknowledgment is not observed within a fixed timeout period.

1.4.3 addressing: *See:* **extended addressing (32-bit); extended addressing (64-bit); fixed addressing (64-bit).**

1.4.4 agent: An active switch, switch-like component, or bridge, between the requester and responder. During normal system operation, the agent is transparent to the requester and responder.

1.4.5 backplane: A subassembly that holds the connectors into which one or more boards can be plugged. In addition to providing bus signal connections, the backplane usually provides power connections, power status information, and physical position information to the board.

1.4.6 big addressian: A term used to describe the physical location of data-byte addresses on a multiplexed address/data bus. On a big-addressian bus, the data byte with the largest address is multiplexed (in time or space) with the least-significant byte of the address.

1.4.7 big endian: A term used to describe the arithmetic significance of data-byte addresses within a multi-byte register. Within a big-endian register or register set, the data byte with the largest address is the least significant.

1.4.8 board: The physical component that is inserted into one of the backplane connectors.

1.4.9 bridge: A hardware adapter that forwards transactions between buses.

1.4.10 broadcast transaction: A transaction that is distributed to all nodes on a bus.

1.4.11 buffered write: A write transaction that appears to complete when the request is queued in the agent or responder. A buffered-write transaction returns an optimistic (done_correct) status before the responder's completion status (which could report an error) is available.

1.4.12 bus-dependent: A term used to describe parameters that are defined by the bus standards that conform to this standard. Although the CSR Architecture may constrain the definition of these fields, their detailed definition is provided by the appropriate bus standard.

1.4.13 bus standard: An abbreviated notation used throughout this document, rather than the more exact "bus standard document that claims conformance to this specification."

1.4.14 byte: A byte is 8 bits of data.

1.4.15 coherent transaction: A transaction (typically read or write) that provides protocols for checking and maintaining consistency with other caches. Coherent transactions are expected to address a cache-line. For example, tightly coupled multiprocessors are expected to use coherent transactions when accessing shared-memory resident data.

1.4.16 command_reset: An initialization event that is initiated by a write to the RESET_START register.

1.4.17 company_id: A 24-bit binary value used to identify a company within the context of the CSR Architecture. The company_id values are expected to be uniquely assigned to each company.

1.4.18 conflict_error: An error-status code that is returned when a transaction has been transmitted successfully, but a queue or usage conflict inhibits the transaction completion. A conflict_error status is returned to the original requester, which is expected to retry the transaction. This is different than a bus-dependent delay (wait or busy status), which delays the forwarding of a transaction or subaction across the bus.

1.4.19 CSR: An abbreviation for control and status register. A CSR is a quadlet register that is accessed through read4 or write4 transactions and is used to observe a node's state or to control its operation.

1.4.20 CSR Architecture: A term that refers to this International Standard.

1.4.21 dead state: A node state that is reflected by the value of 3 in the STATE_CLEAR.*state* field. A node enters the dead state when a fatal error has been detected and the node is connected but no longer operational. Note that the severest errors could leave the node in a broken state, with its registers undefined, rather than indicating a dead state.

1.4.22 diagnostic test: A test, or collection of tests, that is invoked by writing to the TEST_START register. There are four forms of diagnostic tests: initialization tests, extended tests, manual tests, and system tests.

1.4.23 directory: A contiguous collection of one or more entries, which is contained within the node's ROM.

1.4.24 directory entry: A ROM entry that specifies the address of another ROM directory.

1.4.25 disconnected state: A state in which the node no longer responds to bus transactions. Since the node no longer responds to bus transactions, a power_reset is required to change to another node state.

1.4.26 disruptive test: A test that is invoked through a write to the TEST_START register and disrupts the node's operation by temporarily moving the node to the testing state.

1.4.27 DMA: A direct memory access (or simply DMA) architecture is an optional capability of an I/O controller. After being started by the processor, I/O controllers with DMA capabilities can access their commands, fetch data, and report status by accessing memory directly.

1.4.28 done_correct: A status code that is returned when a transaction is completed without errors. On many buses, the done_correct status is implicitly assumed when no error-status codes are observed.

1.4.29 doublet: Two bytes (16 bits) of data.

1.4.30 emperor processor: The monarch processor that is selected to initialize and configure the system. On a single-bus system, the monarch and emperor processor are always the same. On a multiple-bus system, the single emperor processor is selected from the available monarch processors.