UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Apple Inc.

Petitioner

v.

Papst Licensing GmbH & Co.

Patent Owner

Case No. Unassigned
Patent 8,504,746

DECLARATION OF SCOTT BENNETT, Ph.D.

9 August 2016

I, Scott Bennet, Ph.D., resident of Urbana, Illinois, hereby declare as follows:

Introduction and Qualifications

1. I have been retained by Sterne, Kessler, Goldstein & Fox to provide my opinions concerning the public availability of certain documents at issue in *inter partes* review proceedings for U.S. Patent No. 8,504,746.

2. My curriculum vitae is appended to this document as Appendix A. From 1956 to 1960, I attended Oberlin College, where I received an A.B. in English. I then attended Indiana University, where I received an M.A. in 1966 and a Ph.D. in 1967, both in English. In 1976, I received a M.S. in Library Science from the University of Illinois. I also served at the University of Illinois at Urbana-Champaign in two capacities. First, from 1967 to 1974, I was an Assistant Professor of English; then from 1974 to 1981, I was an Instructor, Assistant Professor, and Associate Professor of Library Science.

3. From 1981 to 1989, I served as the Assistant University Librarian for Collection Management, Northwestern University. From 1989 to 1994, I served as the Director of The Milton S. Eisenhower Library at The Johns Hopkins University. From 1994 to 2001, I served as the University Librarian at Yale University. In 2001, I retired from Yale University.

4. Since then, I have served in multiple capacities for various organizations, including as a consultant on library space planning from 2004 to the present, as a Senior Advisor for the library program of the Council of Independent Colleges from 2001 to 2009, as a member of the Wartburg College Library Advisory Board from 2004 to the present, and as a Visiting Professor at the Graduate School of Library and Information Science, University of Illinois at Urbana-Champaign, in the Fall of 2003. I was a founding partner of Prior Art Documentation Services, LLC, in 2015.

5. Over the course of my work as a librarian, professor, researcher, and author of numerous publications, I have had extensive experience with cataloging and online library management systems built around Machine-Readable Cataloging (MARC) standards. As a consultant, I have substantial experience in authenticating documents and establishing the date when they were available to persons exercising reasonable diligence.

6. In the course of more than fifty years of academic life, I have myself been an active researcher. I have collaborated with many individual researchers and, as a librarian, worked in the services of thousands of researchers at four prominent research universities. Members of my family are university researchers. Over the years, I have read some of the voluminous professional literature on the information seeking behaviors of academic researchers. And as an educator, I

have a broad knowledge of the ways in which students in a variety of disciplines learn to master the bibliographic resources used in their disciplines. In all of these ways, I have a general knowledge of the how researchers work.

7. My work in this matter is being billed at my standard consulting rate of $88 per hour. My compensation is not in any way contingent upon the outcome of this or any other *inter partes* review. I have no financial or personal interest in the outcome of this proceeding or any related litigation.

Scope of this Declaration

8. I am not a lawyer and I am not rendering an opinion on the legal question of whether any particular document is, or is not, a "printed publication" under the law.

9. I am, however, rendering my expert opinion on when and how each of the documents addressed herein was disseminated or otherwise made available to the extent that persons interested and ordinarily skilled in the subject matter or art, exercising reasonable diligence, could have located the documents before 4 March 1997.

10. I reserve the right to supplement my opinion in the future to respond to any arguments that the Patent Owner raises and to take into account new information as it becomes available.

Materials Considered in Forming My Opinion

11. In forming the opinions expressed in this declaration, I have reviewed the documents and attachments referenced below.

Document 1. Schmidt, Friedhelm. The SCSI Bus and IDE Interface: Protocols, Applications and Programming, translated by J. Michael Schultz. Wokingham, England: Addison-Wesley Publishers, 1995.

12. The following Attachments are true and accurate representations of library material and online documents and records, as they are identified below. Unless otherwise indicated, all Attachments are records made in the regular course of business and available to the public. All attachments were created on 29 July – 9 August 2016. Each item is a type of material that experts in my field would reasonably rely upon to in forming their opinions.

Attachment 1a: Statewide Illinois Library Catalog record for Schmidt

Attachment 1b: University of Michigan Library catalog record for Schmidt

Attachment 1c: Copy of Schmidt from a copy provided by counsel

Attachment 1d: Swiss National Library catalog record for Schmidt

Background Information

13. *Persons of ordinary skill in the art*. I am told by counsel that the subject matter of this proceeding relates to data interfacing between a host computer and a peripheral device.

14. I am told by counsel that a person having ordinary skill in the art at the relevant time would have had at least a four-year degree in electrical engineering, computer science, computer engineering, or a related field of study, or equivalent experience, and at least two years' experience in studying or developing computer interfaces or peripherals and storage related software.

15. It is my opinion that such a person would have been engaged in research starting at least in undergraduate school, learning though study and practice in the field and possibly through formal instruction the bibliographic resources relevant to his or her research. In the 1980s and 1990s such a person would have had access to a vast array of long-established print resources in electrical engineering, computer science, and computer engineering as well as to a rich and fast changing set of online resources providing indexing information, abstracts, and full text services for electrical engineering.

16. *Library catalog records*. Libraries world-wide use the MARC format for catalog records; this machine readable format was developed at the Library of Congress in the 1960s.

17. MARC formatted records use numerous tags and code. For example, they provide a variety of subject access points based on the content of the document being cataloged. All may be found in the MARC Fields 6XX. Particularly important are the MARC Field 600, which identifies personal names,

and the MARC Field 650, which identifies topical terms. An ordinarily skilled researcher might discover material relevant to his or her topic by a search using the access points provided in the MARC Fields 6XX.

18. The MARC Field 040, subfield a, identifies the library or other entity that created the original catalog record for a given document and transcribed it into machine readable form. The MARC Field 008 identifies the date when this first catalog record was entered on the file. This date persists in all subsequent uses of the first catalog record, although newly created records for the same document will show a new date.

19. Other MARC Fields useful in establishing the authenticity of a book are MARC Field 020 International Standard Book Number, MARC Field 100 Main Entry, MARC Field 245 Title Statement, MARC Field 260 Publication information, MARC Field 600 Physical Description, and MARC Field 500 General Notes.

20. WorldCat is the world's largest public online catalog, maintained by the Online Computer Library Center, Inc., or OCLC, and built with the records created by the thousands of libraries that are members of OCLC. WorldCat provides a user-friendly interface for the public to use MARC records; it requires no knowledge of MARC tags and codes. WorldCat records appear in many different catalogs, including the Statewide Illinois Library Catalog. The date a

given catalog record was created (corresponding to the MARC Field 008) appears in some detailed WorldCat records as the Date of Entry.

21. The public availability of MARC formatted catalog records and detailed WorldCat records showing the Date of Entry varies.

22. When a book has been cataloged, it will normally be made available to readers soon thereafter—normally within a few days or (at most) within a few weeks of cataloging.

Consideration of individual documents

Document 1. Schmidt, Friedhelm. The SCSI Bus and IDE Interface: Protocols, Applications and Programming, translated by J. Michael Schultz. Wokingham, England: Addison-Wesley Publishers, 1995. Herein referred to as Schmidt.

23. Document 1 is a book written by Friedhelm Schmidt and published by Addison-Wesley in 1995. This book is herein referred to as Schmidt.

24. Based on the evidence presented below—a book held in numerous libraries and library cataloging—it is my opinion that Schmidt was publicly available from the publisher by the book's publication in 1995; and that Schmidt was bibliographically discoverable by an ordinarily skilled researcher by mid-June 1995 and available in at least one library by late November 1995.

25. Attachment 1a is a true and accurate copy of the Statewide Illinois Library Catalog showing that Schmidt is held by 127 libraries world-wide. An ordinarily skilled researcher would have no difficulty locating copies of this book.

26. The University of Michigan Library is one library holding Schmidt. Attachment 1b is a true and accurate copy of that library's catalog record, in MARC format, for Schmidt. The MARC Field 650 entries indicate the subject terms under which ordinarily skilled researchers would have found Schmidt.

27. Attachment 1c is a true and accurate copy of the cover, title page, title page verso, preface, table of contents, and Part 1 of Schmidt from a copy of the book provided by counsel. This copy conforms exactly to the cataloging information provided in several key MARC Fields, as an authentic copy of the book would. Specifically, the following MARC Field information as recorded in Attachment 1b is evident in Attachment 1c:

- MARC Field 020 International Standard Book Number: the copy of Schmidt used for this declaration has the ISBN 0-201-42284-0.
- MARC Field 100 Main Entry—Personal Name: in the copy of Schmidt used for this declaration, the author's name is Friedhelm Schmidt.
- MARC Field 245 Title Statement: in the copy of Schmidt used for this declaration, the book's title is The SCSI Bus and IDE Interface. The remainder of the title is Protocols, Applications and Programming.
- MARC Field 260 Publication, Distribution, Etc: in the copy of Schmidt used for this declaration, the places of publication include Wokingham, England and Reading, Massachusetts; Addison-Wesley is the name of the

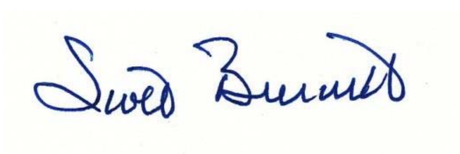
publisher; 1995 is the date of publication; and there is a computer disk in a pocket in the inside back cover. This disc bears the title of Schmidt and the author's name and a December 1994 date.

- MARC Field 600 Physical Description: the copy of Schmidt used for this declaration has six preliminary pages (the table of contents) and 301 pages of text (including the index); it has illustrations; and it is 24 cm. high.
- MARC Field 500 General Note: the copy of Schmidt used for this declaration contains an index.

28. The Swiss National Library is another library that holds Schmidt. Attachment 1d is a true and accurate copy of that library's catalog record, in MARC format, for Schmidt. The MARC Field 008 Field shows this record was created on 7 November 1995 and that Schmidt was publicly available in a least one library soon after this date.

Attestation

29. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statement were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statement may jeopardize the validity of the application or any patent issued thereon.



9 August 2016

Scott Bennett, Ph.D.
Managing Partner
Prior Art Documentation Services LLC

Date

EXHIBIT A: RESUME

SCOTT BENNETT
Yale University Librarian Emeritus

711 South Race
Urbana, Illinois 61801-4132
2scottb@prairienet.org
217-367-9896

EMPLOYMENT

Retired, 2001. Retirement activities include:

- Managing Partner in Prior Art Documentation Services, LLC, 2015-. This firm provides documentation services to patent attorneys; more information is available at http://www.priorartdocumentation.com
- Consultant on library space design, 2004- . This consulting practice is rooted in a research, publication, and public speaking program conducted since I retired from Yale University in 2001. I have served more than 50 colleges and universities in the United States and abroad with projects ranging in likely cost from under $50,000 to over $100 million. More information is available at http://www.libraryspaceplanning.com/
- Senior Advisor for the library program of the **Council of Independent Colleges**, 2001-2009
- Member of the Wartburg College Library Advisory Board, 2004-
- Visiting Professor, Graduate School of Library and Information Science, **University of Illinois at Urbana-Champaign**, Fall 2003

University Librarian, **Yale University**, 1994-2001

Director, The Milton S. Eisenhower Library, **The Johns Hopkins University**, Baltimore, Maryland, 1989-1994

Assistant University Librarian for Collection Management, **Northwestern University**, Evanston, Illinois, 1981-1989

Instructor, Assistant and Associate Professor of Library Administration**, University of Illinois at Urbana-Champaign**, 1974-1981

Assistant Professor of English, **University of Illinois at Urbana-Champaign**, 1967-1974

Woodrow Wilson Teaching Intern, **St. Paul's College**, Lawrenceville, Virginia, 1964-1965

EDUCATION

University of Illinois, M.S., 1976 (Library Science)
Indiana University, M.A., 1966; Ph.D., 1967 (English)
Oberlin College, A.B. magna cum laude, 1960 (English)

HONORS AND AWARDS

Morningside College (Sioux City, IA) Doctor of Humane Letters, 2010

American Council of Learned Societies Fellowship, 1978-1979; Honorary Visiting Research Fellow, Victorian Studies Centre**, University of Leicester**, 1979; **University of Illinois** Summer Faculty Fellowship, 1969

Indiana University Dissertation Year Fellowship and an **Oberlin College** Haskell Fellowship, 1966-1967; **Woodrow Wilson** National Fellow, 1960-1961

PROFESSIONAL ACTIVITIES

American Association for the Advancement of Science: Project on Intellectual Property and Electronic Publishing in Science, 1999-2001

American Association of University Professors: University of Illinois at Urbana-Champaign Chapter Secretary and President, 1975-1978; Illinois Conference Vice President and President, 1978-1984; national Council, 1982-1985, Committee F, 1982-1986, Assembly of State Conferences Executive Committee, 1983-1986, and Committee H, 1997-2001 ; Northwestern University Chapter Secretary/Treasurer, 1985-1986

Association of American Universities: Member of the Research Libraries Task Force on Intellectual Property Rights in an Electronic Environment, 1993-1994, 1995-1996

Association of Research Libraries: Member of the Preservation Committee, 1990-1993; member of the Information Policy Committee, 1993-1995; member of the Working Group on Copyright, 1994-2001; member of the Research Library Leadership and Management Committee, 1999-2001; member of the Board of Directors, 1998-2000

Carnegie Mellon University: Member of the University Libraries Advisory Board, 1994

Center for Research Libraries: Program Committee, 1998-2000

Johns Hopkins University Press: Ex-officio member of the Editorial Board, 1990-1994; Co-director of Project Muse, 1994

Library Administration and Management Association, Public Relations Section, Friends of the Library Committee, 1977-1978

Oberlin College: Member of the Library Visiting Committee, 1990, and of the Steering Committee for the library's capital campaign, 1992-1993; President of the Library Friends, 1992-1993, 2004-2005; member, Friends of the Library Council, 2003-

Research Society for Victorian Periodicals: Executive Board, 1971-1983; Co-chairperson of the Executive Committee on Serials Bibliography, 1976-1982; President, 1977-1982

A Selected Edition of W.D. Howells (one of several editions sponsored by the MLA Center for Editions of American Authors): Associate Textual Editor, 1965-1970; Center for Editions of American Authors panel of textual experts, 1968-1970

Victorian Studies: Editorial Assistant and Managing Editor, 1962-1964

Wartburg College: member, National Advisory Board for the Vogel Library, 2004-

Some other activities: Member of the **Illinois State Library** Statewide Library and Archival Preservation Advisory Panel; member of the **Illinois State Archives** Advisory Board; member of a committee advising the **Illinois Board of Higher Education** on the cooperative management of research collections; chair of a major collaborative research project conducted by the **Research Libraries Group** with support from Conoco, Inc.; active advisor on behalf of the **Illinois Conference AAUP** to faculty and administrators on academic freedom and tenure matters in northern Illinois.

Delegate to **Maryland Governor's Conference on Libraries and Information Service**; principal in initiating state-wide preservation planning in Maryland; principal in an effort to widen the use of mass deacidification for the preservation of library materials through cooperative action by the **Association of Research Libraries** and the **Committee on Institutional Cooperation**; co-instigator of a campus-wide information service for **Johns Hopkins University**; initiated efforts with the **Enoch Pratt Free Library** to provide information services to Baltimore's Empowerment Zones; speaker or panelist on academic publishing, copyright, scholarly communication, national and regional preservation planning, mass deacidification.

Consultant for the **University of British Columbia** (1995), **Princeton University** (1996), **Modern Language Association**, (1995, 1996), **Library of Congress** (1997), **Center for Jewish History** (1998, 2000-), **National Research Council** (1998); Board of Directors for the **Digital Library Federation**, 1996-2001; accreditation visiting team at **Brandeis University** (1997); mentor for **Northern Exposure to Leadership** (1997); instructor and mentor for ARL's **Leadership and Career Development Program** (1999-2000)

At the **Northwestern University Library**, led in the creation of a preservation department and in the renovation of the renovation, for preservation purposes, of the Deering Library book stacks.

At the **Milton S. Eisenhower Library**, led the refocusing and vitalization of client-centered services; strategic planning and organizational restructuring for the library; building renovation planning. Successfully completed a $5 million endowment campaign for the humanities collections and launched a $27 million capital campaign for the library.

At the **Yale University Library**, participated widely in campus-space planning, university budget planning, information technology development, and the promotion of effective teaching and learning; for the library has exercised leadership in space planning and renovation, retrospective conversion of the card catalog, preservation, organizational development, recruitment of minority librarians, intellectual property and copyright issues, scholarly communication, document delivery services among libraries, and instruction in the use of information resources. Oversaw approximately $70 million of library space renovation and construction. Was co-principal investigator for a grant to plan a digital archive for Elsevier Science.

Numerous to invitations speak at regional, national, and other professional meetings and at alumni meetings. Lectured and presented a series of seminars on library management at the **Yunnan University Library**, 2002. Participated in the 2005 International Roundtable for Library and Information Science sponsored by the **Kanazawa Institute of Technology** Library Center and the Council on Library and Information Resources.

PUBLICATIONS

"Putting Learning into Library Planning," *portal: Libraries and the Academy*, 15, 2 (April 2015), 215-231.

"How librarians (and others!) love silos: Three stories from the field " available at the Learning Spaces Collaborary Web site, http://www.pkallsc.org/

"Learning Behaviors and Learning Spaces," *portal: Libraries and the Academy,* 11, 3 (July 2011), 765-789.

"Libraries and Learning: A History of Paradigm Change," *portal: Libraries and the Academy,* 9, 2 (April 2009), 181-197. Judged as the best article published in the 2009 volume of *portal*.

"The Information or the Learning Commons: Which Will We Have?" *Journal of Academic Librarianship*, 34 (May 2008), 183-185. One of the ten most-cited articles published in JAL, 2007-2011.

"Designing for Uncertainty: Three Approaches," *Journal of Academic Librarianship*, 33 (2007), 165–179.

"Campus Cultures Fostering Information Literacy," *portal: Libraries and the Academy*, 7 (2007), 147-167. Included in Library Instruction Round Table Top Twenty library instruction articles published in 2007

"Designing for Uncertainty: Three Approaches," *Journal of Academic Librarianship*, 33 (2007), 165–179.

"First Questions for Designing Higher Education Learning Spaces," *Journal of Academic Librarianship*, 33 (2007), 14-26.

"The Choice for Learning," *Journal of Academic Librarianship*, 32 (2006), 3-13.

With Richard A. O'Connor, "The Power of Place in Learning," *Planning for Higher Education*, 33 (June-August 2005), 28-30

"Righting the Balance," in *Library as Place: Rethinking Roles, Rethinking Space* (Washington, DC: Council on Library and Information Resources, 2005), pp. 10-24

Libraries Designed for Learning (Washington, DC: Council on Library and Information Resources, 2003)

"The Golden Age of Libraries," in *Proceedings of the International Conference on Academic Librarianship in the New Millennium: Roles, Trends, and Global Collaboration*, ed. Haipeng Li (Kunming: Yunnan University Press, 2002), pp. 13-21. This is a slightly different version of the following item.

"The Golden Age of Libraries," *Journal of Academic Librarianship*, 24 (2001), 256-258

"Second Chances. An address . . . at the annual dinner of the Friends of the Oberlin College Library November 13 1999," Friends of the Oberlin College Library, February 2000

"Authors' Rights," *The Journal of Electronic Publishing* (December 1999), http://www.press.umich.edu/jep/05-02/bennett.html

"Information-Based Productivity," in *Technology and Scholarly Communication*, ed. Richard Ekman and Richard E. Quandt (Berkeley, 1999), pp. 73-94

"Just-In-Time Scholarly Monographs: or, Is There a Cavalry Bugle Call for Beleaguered Authors and Publishers?" *The Journal of Electronic Publishing* (September 1998), http://www.press.umich.edu/jep/04-01/bennett.html

"Re-engineering Scholarly Communication: Thoughts Addressed to Authors," *Scholarly Publishing*, 27 (1996), 185-196

"The Copyright Challenge: Strengthening the Public Interest in the Digital Age," *Library Journal*, 15 November 1994, pp. 34-37

"The Management of Intellectual Property," *Computers in Libraries*, 14 (May 1994), 18-20

"Repositioning University Presses in Scholarly Communication," *Journal of Scholarly Publishing*, 25 (1994), 243-248. Reprinted in *The Essential JSP. Critical Insights into the World of Scholarly Publishing. Volume 1: University Presses* (Toronto: University of Toronto Press, 2011), pp. 147-153

"Preservation and the Economic Investment Model," in *Preservation Research and Development. Round Table Proceedings, September 28-29, 1992*, ed. Carrie Beyer (Washington, D.C.: Library of Congress, 1993), pp. 17-18

"Copyright and Innovation in Electronic Publishing: A Commentary," *Journal of Academic Librarianship*, 19 (1993), 87-91; reprinted in condensed form in *Library Issues: Briefings for Faculty and Administrators*, 14 (September 1993)

with Nina Matheson, "Scholarly Articles: Valuable Commodities for Universities," *Chronicle of Higher Education*, 27 May 1992, pp. B1-B3

"Strategies for Increasing [Preservation] Productivity*," Minutes of the [119th] Meeting [of the Association of Research Libraries]* (Washington, D.C., 1992), pp. 39-40

"Management Issues: The Director's Perspective," and "Cooperative Approaches to Mass Deacidification: Mid-Atlantic Region," in *A Roundtable on Mass Deacidification*, ed. Peter G. Sparks (Washington, D.C.: Association of Research Libraries, 1992), pp. 15-18, 54-55

"The Boat that Must Stay Afloat: Academic Libraries in Hard Times," *Scholarly Publishing*, 23 (1992), 131-137

"Buying Time: An Alternative for the Preservation of Library Material," ACLS *Newsletter*, Second Series 3 (Summer, 1991), 10-11

“The Golden Stain of Time: Preserving Victorian Periodicals” in *Investigating Victorian Journalism*, ed. Laurel Brake, Alex Jones, and Lionel Madden (London: Macmillan, 1990), pp. 166-183

“Commentary on the Stephens and Haley Papers” in *Coordinating Cooperative Collection Development: A National Perspective*, an issue of *Resource Sharing and Information Networks*, 2 (1985), 199-201

“The Editorial Character and Readership of *The Penny Magazine*: An Analysis,” *Victorian Periodicals Review*, 17 (1984), 127-141

“Current Initiatives and Issues in Collection Management*,” Journal of Academic Librarianship*, 10 (1984), 257-261; reprinted in *Library Lit: The Best of 85*

“Revolutions in Thought: Serial Publication and the Mass Market for Reading” in *The Victorian Periodical Press: Samplings and Soundings*, ed. Joanne Shattock and Michael Wolff (Leicester: Leicester University Press, 1982), pp. 225-257

“Victorian Newspaper Advertising: Counting What Counts,” *Publishing History*, 8 (1980), 5-18

“Library Friends: A Theoretical History” in *Organizing the Library’s Support: Donors, Volunteers, Friends*, ed. D.W. Krummel, Allerton Park Institute Number 25 (Urbana: University of Illinois Graduate School of Library Science, 1980), pp. 23-32

“The Learned Professor: being a brief account of a scholar [Harris Francis Fletcher] who asked for the Moon, and got it,” *Non Solus*, 7 (1980), 5-12

“Prolegomenon to Serials Bibliography: A Report to the [Research] Society [for Victorian Periodicals],” *Victorian Periodicals Review*, 12 (1979), 3-15

“The Bibliographic Control of Victorian Periodicals” in *Victorian Periodicals: A Guide to Research*, ed. J. Don Vann and Rosemary T. VanArsdel (New York: Modern Language Association, 1978), pp. 21-51

“John Murray’s Family Library and the Cheapening of Books in Early Nineteenth Century Britain,” *Studies in Bibliography*, 29 (1976), 139-166. Reprinted in Stephen Colclough and Alexis Weedon, eds., *The History of the Book in the West: 1800-1914*, Vol. 4 (Farnham, Surrey: Ashgate, 2010), pp. 307-334.

with Robert Carringer, “Dreiser to Sandburg: Three Unpublished Letters,” *Library Chronicle*, 40 (1976), 252-256

“David Douglas and the British Publication of W. D. Howells’ Works,” *Studies in Bibliography*, 25 (1972), 107-124

as primary editor, W. D. Howells, *Indian Summer* (Bloomington: Indiana University Press, 1971)

“The Profession of Authorship: Some Problems for Descriptive Bibliography” in *Research Methods in Librarianship: Historical and Bibliographic Methods in Library Research*, ed. Rolland E. Stevens (Urbana: University of Illinois Graduate School of Library Science, 1971), pp. 74-85 edited with Ronald Gottesman, *Art and Error: Modern Textual Editing* (Bloomington: Indiana University Press, 1970)--also published in London by Methuen, 1970

"Catholic Emancipation, the *Quarterly Review*, and Britain's Constitutional Revolution," *Victorian Studies*, 12 (1969), 283-304

as textual editor, W. D. Howells, *The Altrurian Romances* (Bloomington: Indiana University Press, 1968); introduction and annotation by Clara and Rudolf Kirk

as associate textual editor, W. D. Howells, *Their Wedding Journey* (Bloomington: Indiana University Press, 1968); introduction by John Reeves

"A Concealed Printing in W. D. Howells," *Papers of the Bibliographic Society of America*, 61 (1967), 56-60

editor, *Non Solus*, A Publication of the University of Illinois Library Friends, 1974-1981

editor, Robert B. Downs Publication Fund, University of Illinois Library, 1975-1981

reviews, short articles, etc. in *Victorian Studies*, *Journal of English and German Philology*, *Victorian Periodicals Newsletter*, *Collection Management*, *Nineteenth-Century Literature*, *College & Research Libraries*, *Scholarly Publishing Today*, *ARL Newsletter*, *Serials Review*, *Library Issues*, *S[ociety for] S[cholarly] P[ublishing] Newsletter*, and *Victorian Britain: An Encyclopedia*

Attachment 1a: Statewide Illinois Library Catalog record for Schmidt

Statewide Illinois Library Catalog | UNIV OF ILLINOIS

WorldCat Detailed Record

| Ask A Librarian

- Click on a checkbox to mark a record to be e-mailed or printed in Marked Records.

Home | Databases | Searching | Results — Staff View | My Account | Options | Comments | Exit | Hide tips

List of Records | Detailed Record | Marked Records | Saved Records | Go to page

Subjects Libraries E-mail Bib Print Export Help

WorldCat results for: **(ti: SCSI and ti: Bus) and (ti: IDE and ti: Interface)**. Record **2** of **41**.

Prev 2 Next **Mark:**

Detailed Record | Add/View Comments

The SCSI bus and IDE interface :

protocols, applications, and programming /

Friedhelm **Schmidt**

1995
English Book ix, 301 pages : illustrations ; 24 cm + 1 computer disc (3 1/2 in.)
Wokingham, England ; Reading, Mass. : Addison-Wesley, ; ISBN: 0201422840 9780201422849

Disk includes source code for the program examples listed in the book and a SCSI monitor tool for the testing and troubleshooting SCSI devices.

GET THIS ITEM

Availability: **Check the catalogs in your library.**
- Libraries worldwide that own item: 127
- Search the catalog at the Library of University of Illinois at Urbana-Champaign

External Resources:
- Discover full text Discover UIUC Full Text
- Interlibrary Loan Request
- Cite This Item

FIND RELATED

More Like This: Search for versions with same title and author | Advanced options ...

Title: **The SCSI bus and IDE interface : protocols, applications, and programming /**

Uniform Title: SCSI-Bus und IDE-Schnittstelle. English

Author(s): Schmidt, Friedhelm, 1955-

Publication: Wokingham, England ; Reading, Mass. : Addison-Wesley,

Year: 1995

Description: ix, 301 pages : illustrations ; 24 cm + 1 computer disc (3 1/2 in.)

Language: English

Standard No: **ISBN:** 0201422840; 9780201422849 **LCCN:** 94-23813

Abstract: Disk includes source code for the program examples listed in the book and a SCSI monitor tool for the testing and troubleshooting SCSI devices.

SUBJECT(S)

Descriptor:
SCSI (Computer bus)
IDE (Standard)
Interfaces.
Echange de données informatisées.
IDE (Standard)
SCSI (Computer bus)
Interfaces.
Protocollen (informatica)
Programmeren (computers)
Toepassingen.

Identifier: Interfaces (Computers)

Note(s): Includes index.

Class Descriptors: **LC:** TK7895.B87; **Dewey:** 004.6/2

Responsibility: Friedhelm Schmidt ; translated by J. Michael Schultz.

Vendor Info: Brodart Baker and Taylor YBP Library Services (BROD BTCP YANK) $34.95

Document Type: Book

Entry: 19941021

Update: 20160611

Accession No: **OCLC:** 31518664

Database: WorldCat

Subjects Libraries E-mail Bib Print Export Help

WorldCat results for: **(ti: SCSI and ti: Bus) and (ti: IDE and ti: Interface)**. Record **2** of **41**.

Attachment 1b: University of Michigan Library catalog record for Schmidt

Save to Favorites

The SCSI bus and IDE interface : protocols, applications, and programming / Friedhelm Schmidt ; translated by J. Michael Schultz.

Main Author:	Schmidt, Friedhelm, 1955-
Format:	Book
Language:	English
Published:	1995 Wokingham, England ; Reading, Mass. : Addison-Wesley,

Holdings | Description | Subjects | MARC View

```
LEADER  nam a2200289 a 4500
001     002966639
003     MiU
005     19950526000000.0
006     m d
007     co ug
007     cr bn ---auaua
008     950526s1995 enka 001 0 eng
010           |a 94023813
020           |a 0201422840
035           |a (OCoLC)ocm31518664
040           |a DLC  |c DLC  |d EYM
041     1     |a eng  |h ger
050     0  0  |a TK7895.B87  |b S36 1995
082     0  0  |a 004.6/2  |2 20
100     1     |a Schmidt, Friedhelm,  |d 1955-
240     1  0  |a SCSI-Bus und IDE-Schnittstelle.  |l English
245     1  4  |a The SCSI bus and IDE interface :  |b protocols, applications, and programming /  |c Friedhelm Schmidt ; translated by J. Michael Schultz.
260           |a Wokingham, England ;  |a Reading, Mass. :  |b Addison-Wesley,  |c c1995.
300           |a ix, 301 p. :  |b ill. ;  |c 24 cm. +  |e 1 CD-ROM (4 3/4 in.)
500           |a System requirements for accompanying CD-ROM: IBM-compatible PC; DOS.
500           |a Includes index.
538           |a Mode of access: Internet.
650        0  |a SCSI (Computer bus)
650        0  |a IDE (Standard)
852     0     |a MiU  |b AAEL  |c BKS  |h TK 7895 .B87 S363 1995
970           |a BK  |b Book
971           |a MiU
972           |c 20040625
973           |a HT  |b avail_ht
973           |a AC  |b avail_circ
974           |u mdp.39015033952519  |r ic  |d 20140321  |a 39015033952519  |b AAEL  |c BKS  |f 01
998           |c aut3
999           |a MDP  |b 20  |c 20080802  |l MIU01  |h 0000
```

View Similar Items

The SCSI bus and IDE interface : protocols, applications, and programming /
By: Schmidt, Friedhelm, 1955-
Published in 1998
Format: Book

Small computer system interface (SCSI).
By: United States. National Bureau of Standards.
Published in 1987
Format: Book, Available online

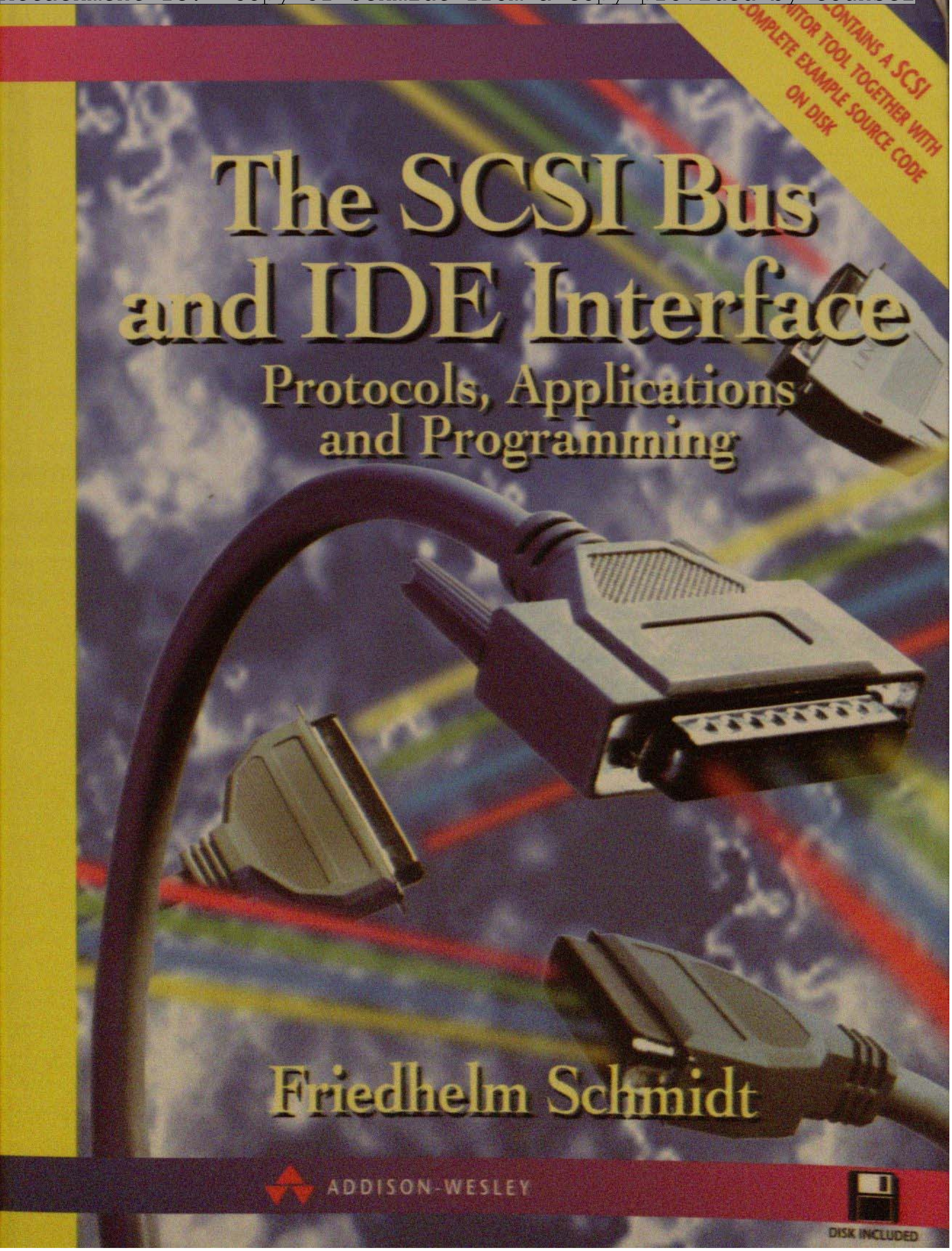
More...

Find Items with Same Subjects

SCSI (Computer bus)
IDE (Standard)

Mirlyn Classic | Mirlyn Mobile

Attachment 1c: Copy of Schmidt from a copy provided by counsel



The SCSI Bus and IDE Interface

Protocols, Applications and Programming

Friedhelm Schmidt

Translated by
J. Michael Schultz
TransTech Translations

ADDISON-WESLEY
PUBLISHING
COMPANY

Wokingham, England • Reading, Massachusetts • Menlo Park, California • New York
Don Mills, Ontario • Amsterdam • Bonn • Sydney • Singapore
Tokyo • Madrid • San Juan • Milan • Paris • Mexico City • Seoul • Taipei

Attachment 1c: Copy of Schmidt from a copy provided by counsel

Translated from the German edition *SCSI-Bus und IDE-Schnittstelle* published by Addison-Wesley (Deutschland) GmbH.

Cover designed by Designers & Partners, Oxford
and printed by The Riverside Printing Co. (Reading) Ltd.
Typeset by VAP Group, Kidlington, Oxon.
Printed and bound in Great Britain at T.J. Press (Padstow) Ltd, Padstow, Cornwall

First printed 1995

ISBN 0-201-42284-0

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library.

Library of Congress Cataloging-in-Publication Data
Schmidt. Friedhelm
(SCSI-Bus und IDE-Schnittstelle. English)
The SCSI bus and IDE interface/Friedhelm Schmidt; translated by Michael Schultz.
p. cm.
Includes index.
ISBN 0-201-42284-0
1. SCSI (Computer bus) 2. IDE (Standard) I. Title
TK7895.B87S36 1995
004. 6'2--dc20

94-23813
CIP

Preface

The SCSI bus and IDE interface are without question the two most important interfaces for computer peripherals in use today. The IDE hard disk interface is found almost exclusively in the world of IBM PC compatibles. The SCSI bus, on the other hand, is designed not only for hard drives but also for tape drives, CD-ROM, scanners, and printers. Almost all modern computers, from PCs to work-stations to mainframes, are equipped with a SCSI interface.

Both SCSI and IDE are ANSI standards. However, aside from the actual ANSI documentation, there exists almost no additional reference material to either specification. The purpose of this book is to fill that void with a clear, concise description of both interfaces. The essential terminology is introduced, while the commands and protocols are broken down in full. In the interest of economy the less important details and options have been omitted in certain cases. Often a specific section in the ANSI documentation will be cited for easy cross-referencing. After reading this book you should be in the position to easily understand relevant technical documentation, including the ANSI specifications themselves.

First and foremost, a thorough introduction to the terminology is in order. Especially with respect to SCSI, there is a deluge of terms and definitions that are used nowhere else or are used differently than in other computer domains. These keywords, which include signal names and interface commands, are typeset in small capital letters, for example FORMAT UNIT.

This book is intended for readers with a broad range of technical backgrounds and interests. Those working on the design of mass storage devices, for example, will find the protocol descriptions extremely useful. Readers writing software or device drivers may have other interests. They will find the hardware descriptions, such as that of the physical organization of a disk drive, very helpful.

This book is not meant to replace the ANSI documentation. On the other hand, those specifications are not meant to explain the technology, rather to define it. It is very difficult to find your way around in the original documentation without an understanding of the subject matter. The book's thorough, in-depth descriptions, along with index and glossary, make it the perfect tutor for IDE and SCSI, as well as a helpful guide to the ANSI literature.

Friedhelm Schmidt
February 1993

Contents

App
App
App
App
Ap

Attachment 1c: Copy of Schmidt from a copy provided by counsel

Part I
Introduction

Attachment 1c: Copy of Schmidt from a copy provided by counsel

1 Computers and peripherals

A computer can be broken down into a number of interdependent functional blocks. The most important of these are the central processing unit (CPU), main memory, input/output (I/O) and mass storage. The CPU executes the instructions of a program, which, along with the necessary data, must reside in main memory at execution time. Therefore, before a program can be run it must be loaded into main memory from mass storage. The data to be processed by the program comes either from mass storage or from an input device such as the keyboard. The CPU accesses memory at least once for each program step in order to read the corresponding machine instructions. In fact, several accesses are usually necessary to read and write data. For this reason the CPU and memory are very tightly coupled: access is uncomplicated and, above all, fast.

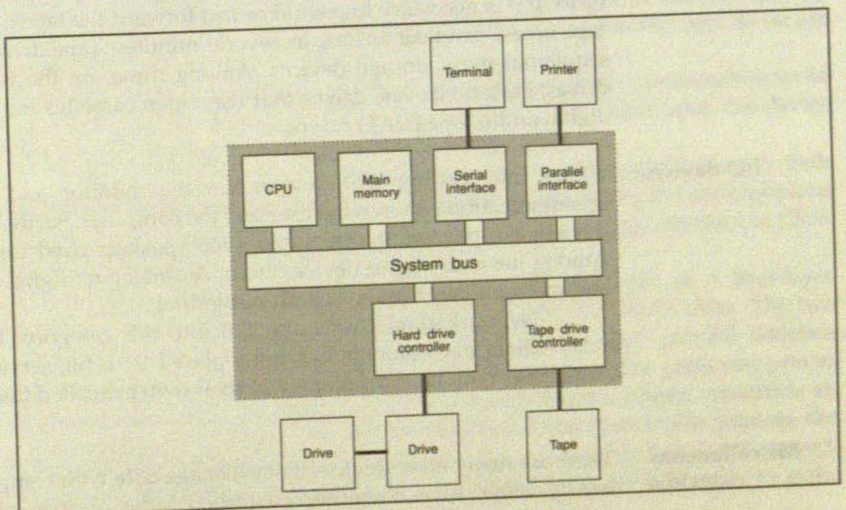


Figure 1.1 Computer system with peripheral devices.

In contrast to memory, I/O devices and mass storage are located further from the CPU, hence the name 'peripherals' (Figure 1.1). Access to such devices is slower and more complicated. Communication with the peripherals is accomplished using an interface such as SCSI or IDE. On the other end of the interface is a controller, which in turn communicates with the CPU and memory.

1.1 Mass storage

A mass storage device is capable of storing data many times the size of main memory. In addition, information stored here is nonvolatile: when the device is turned off the data remains intact.

Hard disks Disk drives or hard disks store information by writing it onto rotating disks. The information is divided up into blocks of fixed length, each of which can be accessed relatively quickly, typically around 30 milliseconds (ms). For this reason hard disks are also referred to as random access mass storage devices. Among the different types of mass storage devices are hard disks, exchangeable medium drives, diskettes, optical disks and CD-ROM.

Tape devices In contrast to hard disks, tape devices (or tape drives) write data sequentially onto magnetic tape. The length of time needed to access a specific block of information depends on which position is presently underneath the read/write head. If it is necessary to rewind or fast forward the tape a very long distance, a tape access can take as long as several minutes. Tape drives are also known as sequential mass storage devices. Among these are the traditional reel-to-reel drives, cassette drives, drives that use video cassettes for recording and 4 mm digital audio tape (DAT) drives.

I/O devices Under the heading I/O devices are the monitor and keyboard used for communication between the user and the computer. Further examples of output devices are printers, plotters and even speakers used for outputting speech. Among the many input devices are mice, analog to digital converters, scanners and microphones used in speech recognition.

Network connections also fall into this category. This is especially so today where mass storage is often replaced by a file server across a network. Computers with no mass storage of their own are called diskless workstations.

Miscellaneous devices There are many more devices that exchange data with computers, although one hardly refers to a computer controlled lathe or a music synthesizer as a computer peripheral. Nevertheless, they function as peripherals and communicate with the computer using I/O.

1.2 Peripheral interfaces

Peripheral devices are connected to computer systems via interfaces. The abstract model of a peripheral interface is made up of many layers, the boundaries of which are not always clear, especially for older interfaces. It is also true that some layers are omitted in certain interface definitions. In this book I adhere to a model with four layers for the SCSI interface, as was agreed upon by the American National Standards Institute (ANSI) committee for the first time for SCSI-3. The strata of layers are designed bottom up. All low level layers are mandatory for the implementation of an interface. An uppermost layer, however, can be omitted in some cases. A high level interface refers to the case where all possible levels have been implemented.

Among those things defined in the lowest level are cable and connector types. Also defined are the signal voltages and the current requirements of the drivers. Finally, the timing and coordination of all of the signals of the bus are described here. This lowest level is referred to as the physical interface.

Directly above the physical layer resides the protocol layer. The protocol of an interface contains, for example, information about the difference between data bytes and command bytes and about the exchange of messages between devices. If corrupted data is to be corrected through the use of error correction, this is described in the interface protocol.

On top of the protocol layer lies the peripheral device model. Here the behavior of devices to be connected to the interface is described. These descriptions can be very detailed and precise. The SCSI bus is an example of such a detailed model, where in addition to the characteristics of general purpose SCSI devices, those of hard disks, tape drives, printers and so on are defined.

Finally, some interfaces go so far as to define which commands must be understood by the interface devices. The command set builds upon the device model and represents the fourth layer of the interface.

The term 'interface' always refers to all implemented layers in their entirety. There are distinct peripheral interfaces defined using the same physical level but a unique protocol level. It is also possible for a single interface to allow for different options in the physical level.

The interface used for printers is a good example of a four-layer interface. Figure 1.2 makes the relationships among the layers clear. The two lower levels are covered by the Centronics interface. This parallel interface contains the definition of the physical and protocol layers. The particular printer model in Figure 1.2 is a page printer. This means that the printer constructs an entire page in internal memory before printing it. In contrast to line printers, the lines of a page can be sent in any order as long as a page boundary is not crossed. However, once a page is printed it is impossible to retrieve it in order to make changes.

The page description language PostScript is an excellent example of a large and complex command set. It is built upon the page printer model and makes it possible to output text as well as various graphic elements. These elements can be positioned freely on the current page. Naturally, there are other

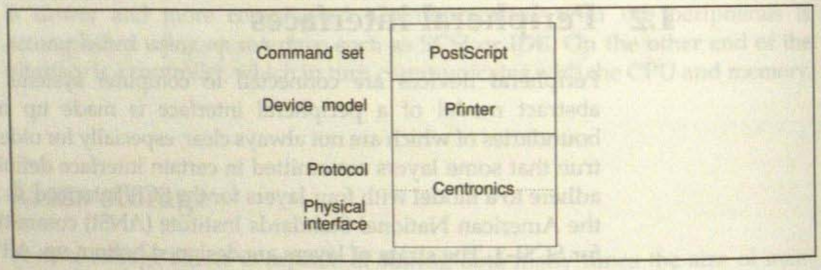


Figure 1.2 Layers of a printer interface.

such page formatting languages written for the page printer model. This makes the division between device and language very intuitive.

As you can see, this interface is complete in that it contains all four interface layers. If you purchase a printer with such an interface, it makes no difference which brand name you choose. As long as it is true to the interface specification it will work with any computer also equipped with the printer interface. However, if you were to omit even only the uppermost layer of the specification, then the interface description would be incomplete. It would still be possible to connect up the printer, but whether it would function properly would be a matter of luck.

The IDE interface and the SCSI bus are likewise complete interface definitions. Before getting to these, however, I would like to introduce in Chapter 2 a few classic examples of peripheral interfaces. For the most part their definitions contain only the lower layers of the interface model. This chapter will help to underscore the difference between traditional interfaces on the one hand and the complete IDE and SCSI interfaces on the other.

2 Traditional peripheral interfaces

This chapter will help to familiarize you with several classic peripheral interfaces of the computer industry. As with the printer interface outlined in Chapter 1, these will be described within the framework of the layered interface model. These descriptions are by no means comprehensive; complete specifications would turn this book into several volumes.

I have two goals in mind in presenting these interfaces. First of all, the interfaces are very simple; they will allow you to become acquainted with interface characteristics that are valid for all interfaces, including computer buses. Secondly, to a certain degree these specifications are the forerunners of competition to the IDE and SCSI bus interfaces. A background in the more traditional interfaces will make it much easier to evaluate and understand their modern descendants, the main topic of this book.

2.1 The RS-232 serial interface

RS-232C is the most widely used serial interface. 'Serial' means that the data is transferred one bit at a time across a single connection. RS-232C is used mainly for the connection of computer terminals and printers. Nonetheless, it is also appropriate for the exchange of data between computers. Machine tools and measurement instruments are frequently connected to computers using RS-232C. Understandably, it is not a device specific interface. RS-232C is the responsibility of the Electronic Industries Association (EIA).

The specification for RS-232C contains the physical layer and hardware protocol. In addition, there are software protocols, of which only a few build on top of the RS-232 hardware protocol. This leads to an uncommon situation with RS-232C and other serial interfaces – not all applications use all of the signals. Frequently cables are used that conduct only a few of the defined signals, a situation that would be unthinkable for IDE or SCSI. I concentrate here on a variation of the interface using only three signals, which I call mini-RS-232.

The physical interface Mini-RS-232 establishes a bidirectional point-to-point connection between equipment. Each direction has its own data signal and a single ground signal is shared. The data signals are called TD (transmit data) and RD (receive data). When two devices are coupled to each other, these signals are crossed such that the TD of one device connects to the RD of the other (Figure 2.1).

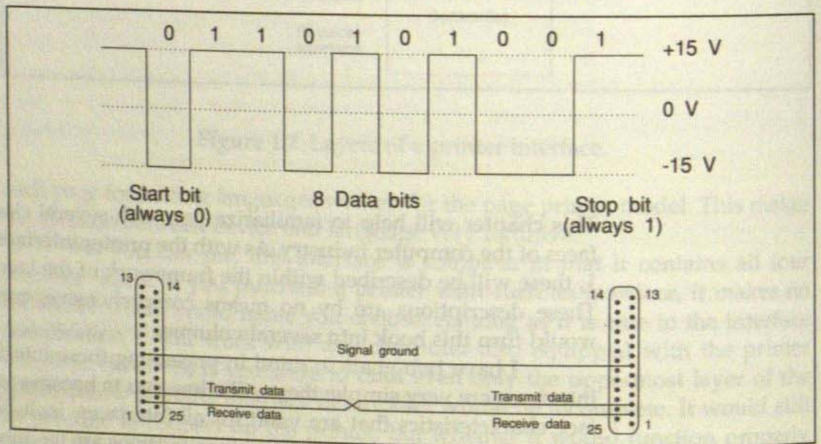


Figure 2.1 Physical interface: mini-RS-232.

The connector chosen by the EIA standard is the 25-pin DB25. Other connectors, however, are frequently employed, such as the DB9 for the IBM AT or the RJ11 telephone connector used in various minicomputers.

On the signal lines, a logical 1 is represented by a voltage between +5 V and +15 V, and the receiver recognizes anything above +3 V as such. Likewise, logical 0 is represented by a signal voltage between –5 V and –15 V. Again, the receiver recognizes any signal below –3 V as such.

Data transfer takes place serially, character by character. The characters are further broken down into bits, which are sent across the line one by one. On the other end, the receiver then assembles the bits back into characters. The number of bits per character lies between five and eight; eight is precisely what is needed to transfer one byte. The data bits are preceded by a start bit and followed by a stop bit. In addition, a parity bit may be sent for error detection. The transfer rate can range between 75 and 115 000 bits per second (baud), and a cable alone cannot compensate for different transfer rates; the devices must be set at the same speed otherwise no exchange of data can take place.

Now comes a rather confusing point: this method of transfer over the serial interface is called asynchronous even though the data is sent and received relative to a clock. Among other serial interfaces the term 'synchronous' is used whenever a clock is involved. For RS-232C, however, the transfer is referred to as asynchronous because the clocks are not tied to each other. The RS-232C specification includes signals that allow the sender and the receiver to use the same clock for data transfer. When these signals are employed the data transfer is referred to as synchronous. True asynchronous transfer uses control signals to exchange data. This point, among others, will be made clear in Section 2.2.

As a rule of thumb, when thinking about data throughput you can consider a byte or character to be 10 bits (one stop, one start and eight data bits). When the fastest transfer rate possible is employed, namely 115 000 bits per second, the maximum throughput is approximately 11.5 Kbytes per second.

The protocol Mini-RS-232 has no protocol of its own. However, there is a protocol that is often used with the interface, called the XON/XOFF protocol (Figure 2.2). It works in the following way. When the receiving device is no longer able to take on data from the sender, it sends a special character, an XOFF byte, to indicate this. Later, when it is ready to continue receiving data, it sends an XON byte to tell the sender to proceed. This protocol is in no way error proof – characters are sometimes lost. In addition, the protocol cannot be used for bidirectional transfer of binary data. The reason for this restriction is simple: for text data only a subset of the possible bytes is sent over the interface, those corresponding to letters, numbers, and symbols. This leaves room for a number of special characters, of which XON and XOFF are examples. When, on the other hand, binary data is transferred, the data is not restricted to certain characters; any binary pattern may occur. In this situation there is no room for the special characters and the XON/XOFF protocol is unusable. For connecting monitors and printers, however, the protocol is actually very practical.

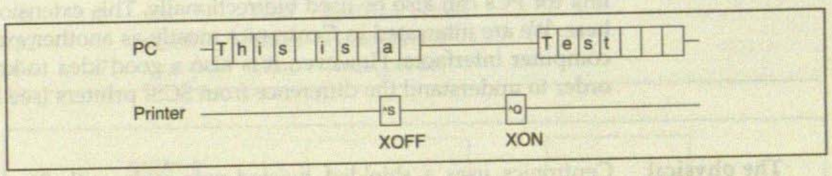


Figure 2.2 XON/XOFF protocol.

An example of a higher level protocol for the transfer of binary data (file transfer) is Kermit. This public domain program can be used at no cost for non-commercial purposes. A number of computer manufacturers have also developed their own internal protocols built on top of RS-232.

Commands There are no commands special to the RS-232 interface. As RS-232 was developed, commands were designed for specific devices apart from the interface. SCSI is among the first interfaces to define universal command sets for whole device classes.

Nevertheless, some command sets have been designed for use with RS-232. Examples are page formatting languages for printers, such as PostScript.

Summary As you can see, an interface that builds on top of RS-232 has many possible variations. The complete description of my printer–PC interface would be: RS-232 at 9600 baud, 1 stop bit, no parity, XON/XOFF protocol, PostScript. If I were to change a parameter for only the printer or only the PC, for example by not sending PostScript or starting to use a parity bit, nothing would print. Although mini-RS-232 appears to be simple (only three wires), there are almost an uncountable number of ways in which the connection can fail. What is missing is a protocol that allows the devices to agree upon the available options. Although RS-232 has given a good portion of frustration to just about everyone who has worked with it, it nonetheless has the decided advantage that it exists on every computer and is also device independent.

2.2 The Centronics printer interface

The Centronics interface is a parallel interface developed for printers. It is an industry standard that, to my knowledge, has never been officially approved. As a result there are many variations. This is especially so with respect to the status signals that reflect the printer's current state. Centronics defines the physical interface and the protocol. As a command set, either PostScript or another printer language is used.

Originally developed as a unidirectional interface, the parallel printer link for PCs can also be used bidirectionally. This extension is not our concern here. We are interested in Centronics mostly as another example of the various computer interfaces. However, it is also a good idea to know this interface in order to understand the difference from SCSI printers (see Figure 2.3).

The physical interface Centronics uses a shielded twisted-pair cable with 36 signals, of maximum length 5 meters (about 16 feet). A 36-pin amphenol connector is used on the printer end, which most people have come to refer to as a Centronics connector. The computer end of the cable has either a corresponding female Centronics or a female DB25.

Electrical specifications The signal voltages correspond to those for transistor–transistor logic (TTL). A 0 is recognized from 0 V to +0.8 V, a 1 from +2.4 V to +5.0 V. Table 2.1 lists the signals of the Centronics interface. Note that I have described the data signals starting with 0; that is, using the logical names. The actual signal names, however, are DATA1 to DATA8.

Data transfer takes place in parallel across signals DATA1 to DATA8. The signals STROBE, BUSY and ACKNLG control the sequencing, which is shown in Figure 2.3. The term 'protocol' does not apply completely here. Relative to our layer model, this timing belongs to the definition of the physical interface.

Table 2.1 The signals of the Centronics interface.

Pin (Cen)	*Pin (DB25)*	*Signal*	*Source*	*Description*
1	1	STROBE	Host	Indicates valid data on DATA1–8
2	2	DATA1	Host	Data bit 0
3	3	DATA2	Host	Data bit 1
4	4	DATA3	Host	Data bit 2
5	5	DATA4	Host	Data bit 3
6	6	DATA5	Host	Data bit 4
7	7	DATA6	Host	Data bit 5
7	8	DATA7	Host	Data bit 6
9	9	DATA8	Host	Data bit 7
10	10	ACKNLG	Printer	Indicates printer has accepted DATA1–8
11	11	BUSY	Printer	Indicates printer is not ready for new data
12	12	PE	Printer	Paper error
13	13	SELECT	Printer	Printer is online
14	14	AUTOFEED	Host	The printer should add a carriage return to each line feed
16		SIGNAL GROUND		
17		CHASSIS GROUND		
18		+5V	Printer	+5 V power (50 mA maximum)
19–30	18–25	SIGNAL GROUND		
31	16	INIT	Host	Initialize printer
32	15	ERROR	Printer	General error
36	17	SLCT IN	Host	Select printer

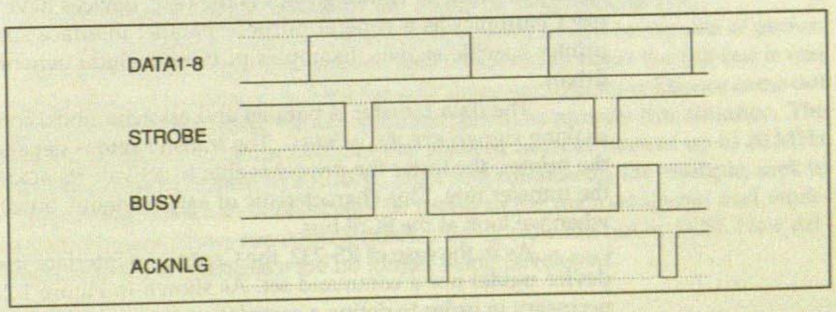


Figure 2.3 Centronics interface timing.

Request/ acknowledge handshake

The transfer of a byte begins when the computer sets the 8 bits on signals DATA1 to DATA8. After waiting for at least a microsecond, it then activates a pulse across STROBE, which indicates that there is valid data on the data lines. In response, the printer sets BUSY and reads the data byte. As soon as the byte has been successfully read and the printer is ready to receive the next byte, it clears the BUSY signal and sends a pulse across the ACKNLG line. Now the computer may change the data signals and send the next STROBE for the next byte. This method of data transfer, where a signal is used to indicate a request (here STROBE) and another to acknowledge that request (here ACKNLG), is called asynchronous. The mechanism itself is termed request/acknowledge handshake.

Throughput Throughput, or the amount of data transferred per second, is dependent upon how long the printer leaves its BUSY signal active for each byte. The other signals involved in the handshake need at least 4 microseconds (μs) in total. If a printer were exceptionally fast, it could accept a byte in around 10 μs. This would correspond to a data rate of 100 Kbytes per second. The handbook for my laser printer reports a value of approximately 100 μs for the length of BUSY, which allows for a rate no faster than 10 Kbytes per second.

The protocol The Centronics interface protocol is very simple. The flow of data is solely the responsibility of the physical layer. When the printer is not able to receive data it simply holds BUSY active. There are, however, a couple of status signals that reflect the printer's status. These fall under the category of message exchange, which places them in the protocol layer. These signals are PE, SELECT, and ERROR. In addition to these are the control signals AUTOFEED, INIT, and SLCT IN. All of these signals are described in Table 2.1.

Summary The Centronics printer interface is our first example of a device specific interface. The method of data transfer is very similar to many parallel interfaces. Nevertheless, the status signals for end of paper and carriage return pertain strictly to printers. Although this is the case, devices have been developed that use Centronics as a general purpose parallel interface simply by ignoring the printer specific signals. Examples of these include network adapters and disk drives.

The data transfer is parallel and asynchronous, controlled by the handshaking signals STROBE/ACKNLG. The transfer rate is dependent on the speed of the printer: the faster the printer is able to activate its ACKNLG signal, the higher the transfer rate. This characteristic of asynchronous transfer will appear again when we look at the SCSI bus.

As in the case of RS-232, the Centronics interface itself contains neither a device model nor a command set. As shown in Figure 1.2, all components are necessary in order to define a complete printer interface. On the other hand, the interface as it stands is flexible. There are even tape back-up devices that take advantage of this very adaptable interface.

Centronics, like RS-232, establishes a point-to-point connection between devices. This means that only a single printer can be used for each interface because the ability to address different devices is lacking. This new feature belongs to the next interface we will discuss.

2.3 Hard disks and their interfaces

This section and the following two sections on ST506 and ESDI delve more deeply into details than previous sections, because it is here that the foundation for understanding IDE and SCSI is laid. If you are not well acquainted with the internals and workings of hard disks, you will find this section especially interesting. Here, you will learn the terminology of the disk drive domain.

A little history

Disk drive interfaces were standardized early on. Beginning in 1975, drives with a diameter of 14 inches and then 8 inches were shipped with the SMD interface. The name comes from the Storage Module Drives of the company, CDC. CDC has since sold its drive production to Seagate. During the late 1980s, as a result of steady improvements, SMD became the favorite interface for 8 inch high performance drives. SMD-E, the final version, had a transfer rate of 24 MHz or about 3 Mbytes per second. The interface, however, could not survive the transition to 5¼ inch drives, primarily because of the very wide cable. As a result SMD died along with 8 inch drives in about 1990.

Five years after the arrival of SMD, Seagate introduced a 5¼ inch drive with a storage capacity of 5 Mbytes. This economical disk drive, at the lower end of the performance scale, used a new interface called ST506. You will often hear ST506/ST412 being used to refer to the same interface. ST506 was not developed from scratch, but evolved from the floppy interface. The transfer rate was increased to 5 MHz (about 625 Kbytes per second) but the method of moving the heads by sending step pulses remained the same. In the past few years, advances have allowed the transfer rate to be doubled once again. However, the demands of modern PCs have finally exceeded the interface's capabilities: ST506 has been steadily losing ground to IDE and SCSI since around 1991.

It was apparent early on that 5¼ inch drives would be capable of performance that ST506 could not support. SMD could have fitted the bill but it was too big and too expensive. In 1983 the disk drive manufacturer Maxtor came out with the Enhanced Small Device Interface (ESDI) to remedy this situation. The ESDI used the same cables as ST506 but allowed transfer rates of up to 20 MHz (2.4 Mbytes per second). In addition, ESDI had commands, for example, seek to track. Today, ESDI can occasionally be found in the microcomputer and workstation domain. However, it too is quickly being crowded out by SCSI. New drives with the ESDI interface are no longer being developed.

The disk drive model

On our way to understanding IDE we will make two stops to examine its predecessors, the ST506 and ESDI interfaces. Before we do this, however, we need to become acquainted with the basic model of a disk drive. A hard disk drive stores information on a set of rotating disks. The information can be written and read any number of times and the data remains intact even after the drive is turned off.

The term 'hard disk' most often refers to a drive with nonremovable media although some removable media drives do use hard disks. A hard disk contrasts with the flexible media used in floppy drives.

Attachment 1c: Copy of Schmidt from a copy provided by counsel

This model of a disk drive will say nothing about the exact method of writing to the medium. This means that it will be valid for magnetic disk drives as well as magneto-optical, diskettes, and removable media drives. CD-ROM and WORM drives, however, do not fall into this category; these formats lack the ability to rewrite information.

Organization of the medium The disk assembly of a drive usually consists of a number of writable surfaces, each of which stores data on concentric rings called tracks. The tracks are further divided into sectors, which are the smallest readable/writable unit. A sector is accessed by first positioning the read/write head above the proper track. The drive then waits until the desired sector rotates underneath the head and reads the data. Writing and reading the sector is done serially bit by bit.

A drive usually contains somewhere between two and eight disks, and both sides of a disk can be utilized for storage. Each surface has its own read/write head although only one track can be written to or read at a given time. The heads are positioned collectively over the tracks. A set of tracks that can be accessed by the heads from a single position is called a cylinder. A consequence of this organization is that every sector of the drive can be uniquely addressed by its cylinder, head and sector numbers. This is referred to as the drive geometry (Figure 2.4).

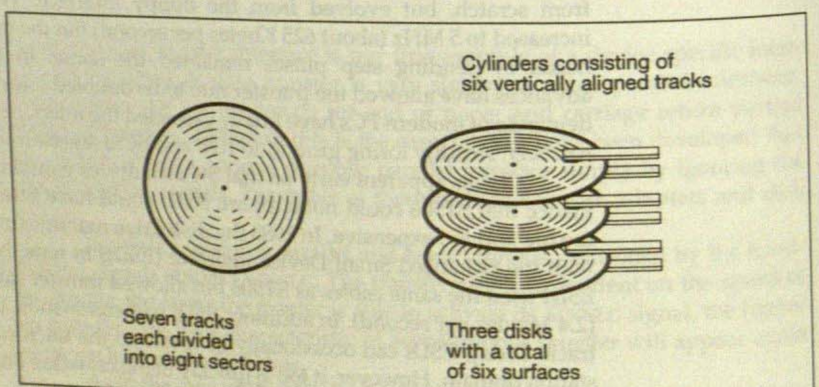


Figure 2.4 Structure of disk medium.

Sector format In order to identify the beginning of a track there is an interface signal called INDEX, which issues a pulse at the precise moment when the heads reach this position. This is where the first sector of the track begins. At the start of the other sectors another interface signal, SECTOR, issues a pulse. If the sector pulse is generated by special circuitry that senses the relative angular position of the disks, the drive is said to be hard sectored. The drive is soft sectored if the beginning of the sector is actually read off the medium by the heads.

A computer uses data in parallel; that is, bytes not bits. The disk formatter is a chip, which in addition to identifying sectors by their sector number also takes the serial data from the heads and groups it properly into bytes. The data separator sits between the heads and the formatter chip. When data is read from the drive it generates an accompanying clock. Finally, the read/write amplifier circuitry amplifies the analog signals to and from the heads. The electronics that pertain to actual reading and writing of information are collectively referred to as the data channel.

A sector is made up of a number of different fields which are together referred to as the sector format. Sector formats differ from interface to interface but a typical format can be described as follows: first comes a field for synchronizing the data separator followed by the address field. The address field contains the cylinder, head, and sector numbers. With this information the controller verifies that it is reading or writing the correct sector. After the address field comes the cyclic redundancy code (CRC) checksum, which is used to check whether the address was read properly. All fields up to this point are collectively referred to as the header. Now comes the data. Here too a synchronization field is used, followed by the actual data of the sector. In the place where the address field has a CRC checksum, the data has a number of error correction code (ECC) bytes. The ECC allows the controller to test whether the data has been correctly written or read. In addition a certain number of incorrectly read bits can actually be corrected using this code. The sector ends with a gap used to even out small differences in motor speed. The number of data bytes in a sector corresponds to its formatted capacity. Typical formatted sector sizes are 512, 1024 and 4096 bytes. The header, ECC and gaps use up space for between 40 and 100 bytes, depending on the sector format (Figure 2.5).

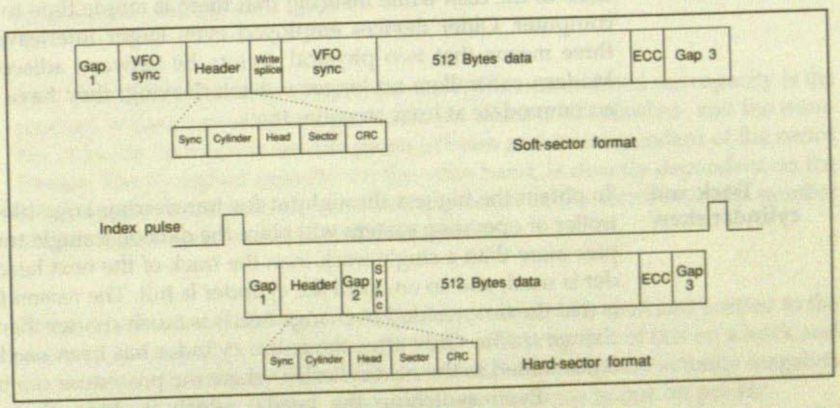


Figure 2.5 Typical sector format.

Formatting, reading, and writing

Only after the drive's medium has been formatted is it usable for data storage. This procedure involves writing not only the headers but also the data field. An arbitrary data pattern is usually written along with the correct ECC. Normally the entire drive is formatted at one time although soft sectoring allows a single track and hard sectoring a single sector to be formatted.

The reading of a sector is relatively simple. As soon as the head is positioned at the correct cylinder, the desired head is chosen and the formatter chip reads headers until the proper address comes by. The data directly following this header is the data required.

Writing a sector is a bit more complicated. A write looks just like a read until the proper header is found, then the amplifier circuitry switches from reading to writing, and the new data, along with ECC, is written. A write-splice is located between the header and the data field to allow time to turn on the write current.

Format characteristics It is not necessarily the case that two sectors with adjacent addresses are adjacent to one another on the medium. The limited throughput of early drive controllers made it necessary to employ certain techniques in the format design. The techniques discussed here are pertinent to IDE as well SCSI.

Interleave Early drive controllers had a very small local buffer which held at most a sector's worth of data. This situation forces the controller to pass the data on to the computer before reading the next sector. If this cannot be accomplished in the time it takes the head to pass over the short gap between sectors, the controller must wait for a complete revolution of the disk for the sector to come around again. For drives of this era, this meant waiting 17 ms for the next sector. In order to avoid this delay, the format of the track can employ an interleave to insure that there is enough time to get ready for the next sector. With a interleave of two, for example, the sector with the next adjacent address is two physical sectors away. This makes it possible to read all sectors of a track with only two rotations of the disk while insuring that there is ample time to pass the data to the computer. Older devices employed even larger interleaves. An interleave of three means that two physical sectors lie between adjacent sector addresses. Modern controllers no longer use interleaving; they have data buffers, which accommodate at least an entire track.

Track and cylinder skew To obtain the highest throughput for transferring large blocks of data the controller or operating system will place the data on a single track. If the data occupies more than a single track then the track of the next head in this same cylinder is used, and so on, until the cylinder is full. The reason for this organization is that the time needed to change heads is much shorter than the time needed to change tracks. Only after the entire cylinder has been used must the heads be repositioned to the next cylinder, where the procedure can begin again.

Even switching the heads, which is done electronically, can cause enough of a delay to miss a sector. When the last sector of a track is read and the heads are switched to begin a new track, the resulting lag may prevent the first sector of the track being read. Waiting for an additional revolution (called 'missing a rev') can be avoided by offsetting the first sector address by one or several physical sectors. This feature is called track skew (spiral offset). Modern controllers, however, are usually capable of a track skew of zero with the help of very fast data channel electronics.

The delay resulting from a seek from one cylinder to the next adjacent cylinder is of the order of 2 ms. In this case as well, an offset can be employed to avoid missing a rev. However, transfers of this size, across cylinder boundaries, rarely occur. Therefore, the implementation of a cylinder skew is often forgone (see Figure 2.6).

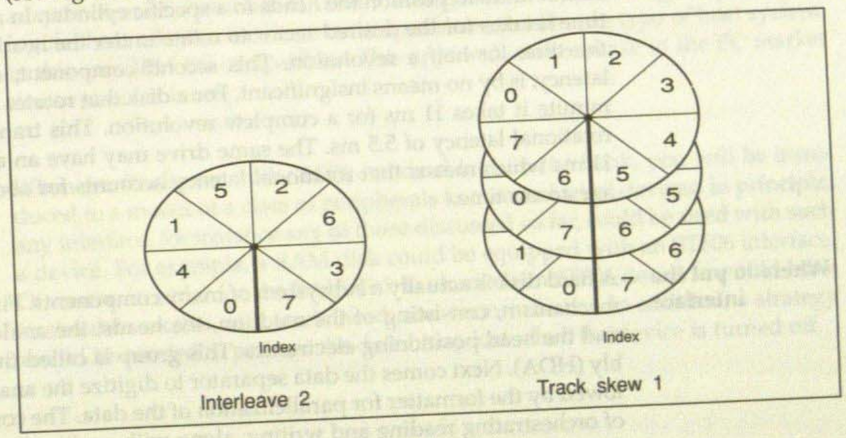


Figure 2.6 Interleave and track skew.

Technical specifications The physical drive model described above is the basis for the technical specifications cited for disk drives. The most important of these are the capacity, transfer rate, and average seek time.

Capacity Two capacities are usually given for a drive. The unformatted net capacity is the product of the number of bits per track, the number of cylinders, and the number of heads. Its value is usually given in bytes and is independent of the sector format. The formatted capacity, on the other hand, is directly dependent on the sector format employed. Its value is the product of the sector size, the number of sectors per track, and the number of heads.

Transfer rate and throughput Transfer rate refers to the speed at which bits are serially read and written to the drive by the heads. It is simply the product of the number of bits on a track and the number of rotations of the disk per second. The units are actually megabits per second, but MHz is often used, which corresponds to one bit per Hz.

Throughput, the amount of data the drive can deliver or accept at the interface on a sustained basis, can be estimated fairly accurately in the following way. Divide the transfer rate by eight (giving the number of bytes per second). Take this result and divide it by the interleave (in this context think of interleave as the number of revolutions needed to read a track). Take off 10% of this value (for headers and so on), and you are left with the approximate throughput of the drive in bytes per second. Throughput, then, is a function of how quickly the medium can be written to and read, plus formatting factors. A drive's peak transfer rate, which is an instantaneous rate, will be higher.

Average access time The average access time has two components. The average seek time is the mean time it takes to position the heads to a specific cylinder. In addition to this is the time it takes for the desired sector to rotate under the heads. On average this is the time for half a revolution. This second component, called the rotational latency, is by no means insignificant. For a disk that rotates at 5400 rotations per minute it takes 11 ms for a complete revolution. This translates to an average rotational latency of 5.5 ms. The same drive may have an average seek time of 11 ms which means that rotational latency accounts for about 30% of the average access time.

Where to put the interface A hard disk is actually a subsystem of many components. First of all is the drive mechanism, consisting of the medium, the heads, the analog data electronics, and the head positioning electronics. This group is called the head disk assembly (HDA). Next comes the data separator to digitize the analog signal data, followed by the formatter for parallelization of the data. The controller is in charge of orchestrating reading and writing, along with positioning the heads. Finally, a host adapter is the link between the controller and the host system (Figure 2.7).

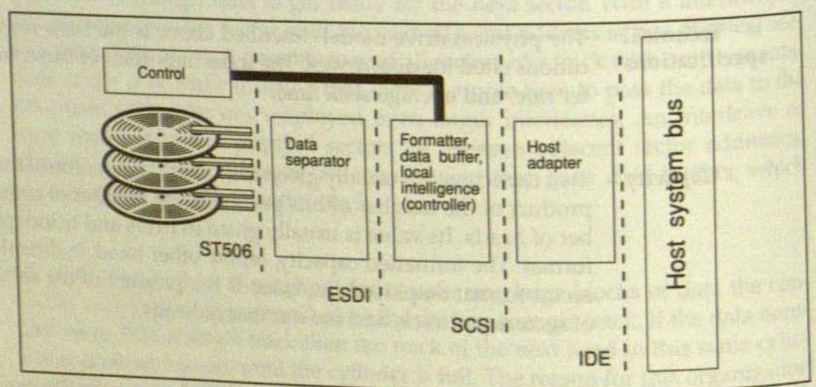


Figure 2.7 Various drive interfaces.

Physically, the interface is the cable that connects the unit built by the drive manufacturer to the computer. There are a number different possible locations along the data channel where this cable can be placed in the design of a drive. The trend, as SCSI's success indicates, is to incorporate more and more functionality in the drive itself. This moves the cable further from the heads, so to speak.

The ST506 interface lies between the analog data electronics and the data separator. One result of this is that the controller determines the analog method of writing data to the drive. In practice, two techniques are employed – modified frequency modulation (MFM) and run length limited (RLL) – across the ST506 interface. The ESDI interface moves one step from ST506 and incorporates the data separator into the drive. Next in line, SCSI packs the formatter and controller into the drive as well. Finally, IDE integrates almost the entire host adapter onto its circuit board. This final step has its disadvantages: by integrating the host adapter, the drive is compatible with only one type of host system, in this case IBM PC compatibles. This approach makes sense in the PC market due to sheer volume.

Summary When we finally reach the SCSI standard later in the book, you will be introduced to a model of a class of peripherals known as logical devices. In principle, any interface, for instance any of those discussed so far, could be used with such a device. For example, a RAM disk could be equipped with an ST506 interface. Of course, in order for the RAM disk to simulate an ST506 device it would have to simulate sectors with track, head, and sector number. In addition, a strategy would be needed to prevent the data being lost when the device is turned off.

2.4 ST506

The ST506 interface lies between the read/write amplifier and the data separator. The data separator is the component that generates a clock and a data signal from the pulses stored on the medium.

Physical interface ST506 can address up to four drives (Figure 2.8). Two cables, named A and B, are used to make the connections. The A cable, which is a single cable, contains control signals, and runs from drive to drive in what is called a daisy chain. The last drive in the chain must contain terminating resistance. The B cable carries the analog read/write data. Each drive has its own B cable. You can recognize a controller that supports four drives by the connectors for a single A cable and four B cables. The maximum cable length for ST506 is 3 meters.

Cables, connectors, and electrical specifications The A cable is a ribbon cable with 34 connections. On the controller end of the cable is a ribbon connector. The two drives are attached using edge connectors. The signals are single ended; 7438/7414 open collector drivers and receivers are used (Figure 2.9).

For the first time, we meet the need for terminating resistors in an interface. The signals of the A cable must be connected to +5 V across a 150 ohm resistor. The resistors for all signals are usually incorporated in a single dual in-line package. Since only the last drive may have termination, terminators are mounted in a socket for easy removal.

Attachment 1c: Copy of Schmidt from a copy provided by counsel

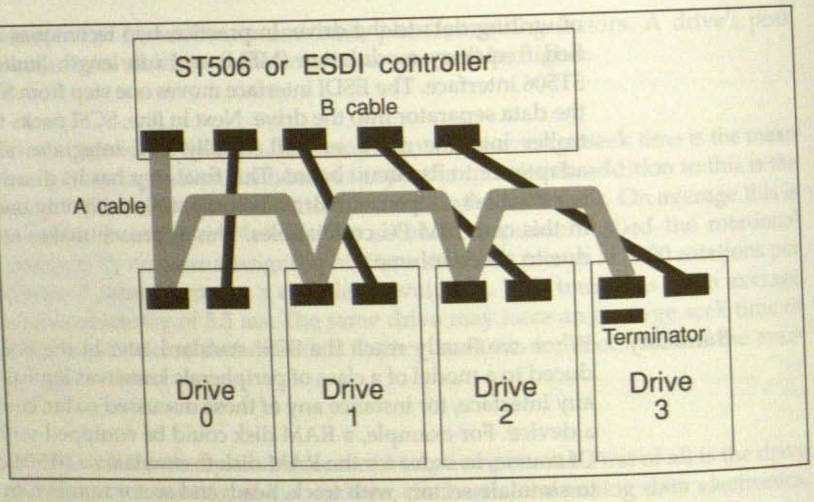


Figure 2.8 ST506 configuration.

The B cable is a ribbon cable with 25 connections. Like the A cable, there is a ribbon cable connector on the controller end and an edge connector on the drive end. The signals here are differential. A 26LS31 and 26LS32 pair is recommended as driver and receiver. Since each drive has its own B cable there is no need to make termination for these signals removable.

Signals Tables 2.2 and 2.3 show the signal assignments for the ST506 cables. Every other signal is ground, which acts as shielding.

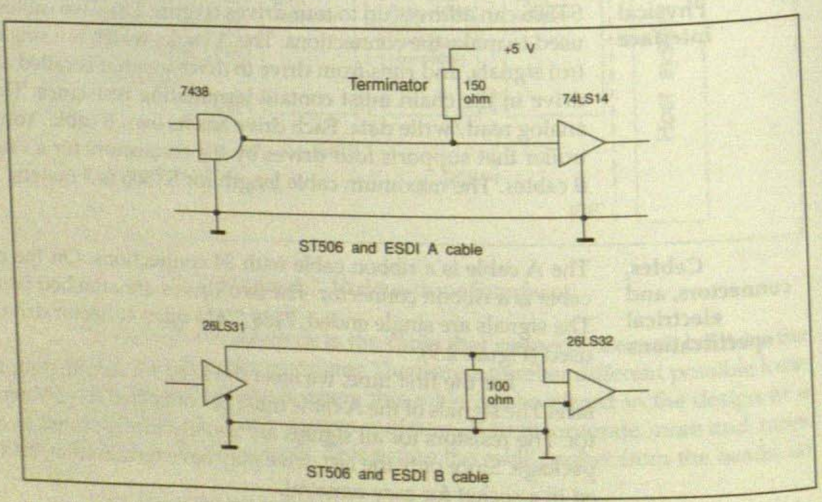


Figure 2.9 ST506 drivers and receivers.

Table 2.2 ST506 A cable signals.

Pin	*Name*	*Signal source*	*Description*
1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33	GROUND		
2	REDUCED WRITE CURRENT/HEAD SELECT 3	Controller	Once used to reduce write current, now bit 3 of head number
4	HEAD SELECT 2	Controller	Bit 2 of head number
6	WRITE GATE	Controller	Activates write current
8	SEEK COMPLETE	Drive	Indicates cylinder has been reached
10	TRACK 00	Drive	Indicates heads are on cylinder zero
12	WRITE FAULT	Controller	Write error
14	HEAD SELECT 0	Controller	Bit 0 of the head number
16	ERROR RECOVERY	Controller	
18	HEAD SELECT 1	Controller	Bit 1 of the head number
20	INDEX	Drive	Indicates beginning of track
22	READY	Drive	The drive is up to speed and ready for read/write
24	STEP	Controller	The heads are to be moved by one cylinder
26	DRIVE SELECT 1	Controller	Drive 1 selected
26	DRIVE SELECT 2	Controller	Drive 2 selected
26	DRIVE SELECT 3	Controller	Drive 3 selected
26	DRIVE SELECT 4	Controller	Drive 4 selected
34	DIRECTION IN	Controller	Selects direction for head movement

Addressing In order to choose a specific sector for reading or writing, the head, cylinder, and sector number of the proper drive must be selected. There are four signals for addressing drives on the ST506 interface labeled DRIVE SELECT 1–4. This means that each drive has a dedicated select line.

In contrast to this, the four signals HEAD SELECT 0–3 select the track under one of 16 possible heads. HEAD SELECT 3 did not exist in the original specification; originally, this connection was used to control the amount of write current. The inner tracks of a disk need less write current than the outer tracks. This signal became unnecessary as disk drives themselves controlled the amount of write current.

The method for choosing a cylinder using the ST506 interface is identical to that for floppy drives. A pulse on the STEP signal causes the heads to move one cylinder in the direction indicated by the signal DIRECTION IN. The status signal SEEK COMPLETE indicates that this positioning of the heads has been completed. Another status signal, TRACK 00, reflects whether or not the heads are on track 0, the outermost cylinder. Using this signal the controller can find track 0 by sending STEP pulses until TRACK 00 is true.

The ST506 interface supports only soft sectoring. For this reason there is no sector pulse among the signals; the desired sector is found by the address

Table 2.3 ST506 B cable signals.

Pin	*Name*	*Signal source*	*Description*
1	DRIVE SELECTED	Drive	Drive is selected
2	GROUND		Ground
3	RESERVED		Reserved
4	GROUND		Ground
5	RESERVED		Reserved
6	GROUND		Ground
7	RESERVED		Reserved
8	GROUND		Ground
9	NOT USED		Not used
10	NOT USED		Not used
11	GROUND		Ground
12	GROUND		Ground
13	+ MFM/RLL WRITE DATA	Controller	Differential write data
14	− MFM/RLL WRITE DATA	Controller	Differential write data
15	GROUND		Ground
16	GROUND		Ground
17	+ MFM/RLL READ DATA	Drive	Differential read data
18	− MFM/RLL READ DATA	Drive	Differential read data
19	GROUND		Ground
20	GROUND		Ground

information in the header. The INDEX signal is generated by the drive and indicates the beginning of the first sector. It is used during formatting to align the sectors of the different heads.

Clearly, an ST506 controller has a lot of responsibility in controlling the drive. The method of positioning the heads is primitive and slow. The only advantage of the step pulse approach is that the number of cylinders is unlimited.

Data encoding

In principle, many methods of data encoding can be used with the ST506 interface. The encoding of the data results in pulses that can be written to the actual drive medium. Originally, MFM encoding was used and more recently RLL encoding. Not all ST506 drives can accommodate RLL, however, because typically a drive's data channel electronics are optimized for MFM.

The data rate for MFM encoding is 5 MHz, which corresponds to 625 Mbytes per second. MFM drives have 17 sectors per track, each of 512 bytes. RLL allows a data rate of 7.5 MHz. Here a track can hold 22 512 byte sectors. Therefore, the use of RLL encoding increases the capacity of the drive by 50%.

Summary

A well-defined protocol layer or command set is not defined for the ST506 interface. The bus timing definitions belong solely to the physical layer. ST506 is undeniably device specific; it makes no sense to use it for anything other than a disk drive.

ST506 has its weak points. The low data transfer rate makes it nearly unusable for higher performance drives. Other low performance characteristics include its lack of commands and step impulse positioning.

Despite its shortcomings the ST506 has been incorporated into systems far beyond the PC domain. Even the IDE and SCSI interfaces show signs of their ST506 origins – you still see, for instance, a parameter to reduce the write current.

2.5 ESDI

ESDI was designed to overcome the deficiencies of ST506. The electrical and mechanical specifications were adopted unchanged from its predecessor. The data separator was moved from the controller to the drive, allowing the maximum data rate to be increased to 20 MHz. Soft as well as hard sectoring of the drive is supported. The interface includes a protocol layer with commands such as seek for head positioning. The drive model for ESDI has been extended to include a format for defect lists. This makes it possible to store in a standard way the list of defects identified by the manufacturer.

The fact that ESDI is also defined for tape drives is not well known and, as far as I know, such a drive has never been built. In this light, ESDI represents a step, however small, toward device independent interfaces. In contrast, SCSI, which appeared at about the same time, was successful in this regard.

Physical interface

ESDI uses the same cable and connectors as ST506. Even the drivers and receivers are the same. The signal assignments, however, are quite different. In addition to the ST506 signals, the A cable includes signals for sending command and status information. Cable A also includes a sector pulse signal used for hard as well as soft sectoring. Drive addressing takes place over three address lines. Although a total of eight addresses are possible, only 1 through 7 are used. Address 0 means that no drive is selected (Table 2.4).

Changes were incorporated into the B cable as well (Table 2.5). Most importantly, the data transferred across the cable is digital. The necessary clock signals were added here, and read and write data is transferred synchronously to these clock signals. Modern ESDI drives have transfer rates of up to 24 MHz or approximately 3 Mbytes per second.

You may have noticed that the index pulse appears on the B cable in addition to the A cable. There is a reason for this. The A cable signal originates only from the selected drive; the B cables, on the other hand, carry the index signals for all drives, selected or not. The controller can use these B cable signals to determine the relative position of each disk. When multiple I/O requests are queued for different drives, the controller can service the request that will result in the shortest rotational latency and seek time. This method of optimization is called rotational position sensing (RPS).

Commands and status/configuration data can be sent across the interface at the same time that data is transferred because dedicated signals for this

Table 2.4 ESDI A cable signals.

Pin	Name	Signal source	Description
1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33	GROUND		Ground
2	HEAD SELECT 3	Controller	Head select bit 3
4	HEAD SELECT 2	Controller	Head select bit 2
6	WRITE GATE	Controller	Turns on write head
8	CONFIG/STATUS DATA	Drive	Status information
10	TRANSFER ACK	Drive	Handshake for serial communication: the drive accepted command bit, or the status bit is valid
12	ATTENTION	Drive	
14	HEAD SELECT 0	Controller	Head select bit 0
16	SECTOR/ADDRESS MARK FOUND	Drive	Sector pulse
18	HEAD SELECT 1	Controller	Head select bit 1
20	INDEX	Drive	Impulse at beginning of track
22	READY	Drive	The drive is ready to receive a command
24	TRANSFER ACK	Controller	Handshake for serial communication: the command bit is valid, or controller expects a status bit
26	DRIVE SELECT 1	Controller	Drive select bit 0
26	DRIVE SELECT 2	Controller	Drive select bit 1
26	DRIVE SELECT 3	Controller	Drive select bit 2
26	COMMAND DATA	Controller	Command transfer
34	DIRECTION IN	Controller	Direction for head movement

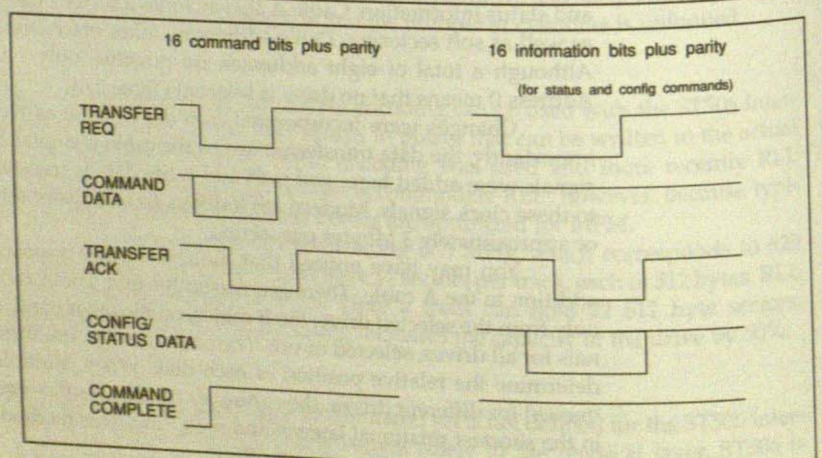


Figure 2.10 ESDI command transfer.

Table 2.5 ESDI B cable signals.

Pin	Name	Signal source	Description
1	DRIVE SELECTED	Drive	Drive is selected
2	N.C.		No correction
3	COMMAND COMPLETE	Drive	Command is finished
4	ADDRESS MARK ENABLE	Controller	
5	RESERVED		Reserved
6	GROUND		Ground
7	+ WRITE CLOCK	Controller	Write clock
8	− WRITE CLOCK	Controller	Write clock
9	RESERVED		Reserved
10	+ READ/REFERENCE CLOCK	Drive	Read clock
11	− READ/REFERENCE CLOCK	Drive	Read clock
12	GROUND		Ground
13	+ NRZ WRITE DATA	Controller	Write data
14	− NRZ WRITE DATA	Controller	Write data
15	GROUND		Ground
16	GROUND		Ground
17	+ NRZ READ DATA	Drive	Read data
18	− NRZ READ DATA	Drive	Read data
19	GROUND		Ground
20	INDEX	Drive	Impulse at beginning of track

purpose reside on the A cable. As before, bus timing details belong to the physical layer of the model.

At this point we turn our attention to a new method of data transfer. Commands and status/configuration data are sent across the interface using asynchronous serial transfers. Four signals are used to support a request/acknowledge handshake – two for data, and two for control. Figure 2.10 shows the timing of these transfers.

Protocol ESDI commands are 16 bits in length plus an additional parity bit. The controller is allowed to send a command when the drive has activated COMMAND COMPLETE. As soon as the first bit is transferred the drive resets COMMAND COMPLETE. Not until after the command has been executed by the drive will it again activate this signal. Some commands request status or configuration data from the drive. This information transfer is part of the command execution. Figure 2.11 shows the sequencing of such a command.

So far the controller has been in charge of initiating activity across the interface, regardless of the direction of the data. If some type of drive error should occur, the drive uses the ATTENTION signal to notify the controller that it has something to say. In response, the controller issues a REQUEST STATUS command to discover the reason for the ATTENTION condition (Figure 2.12).

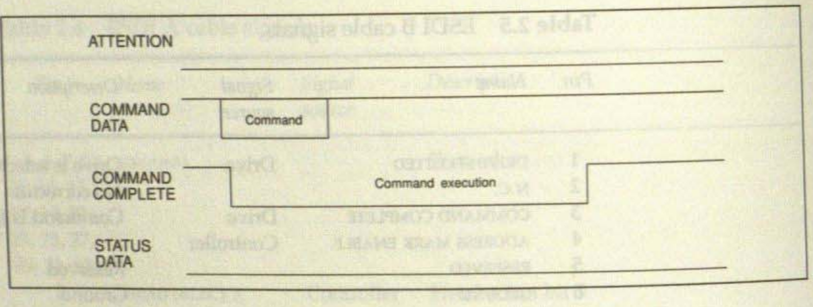


Figure 2.11 Normal command sequence.

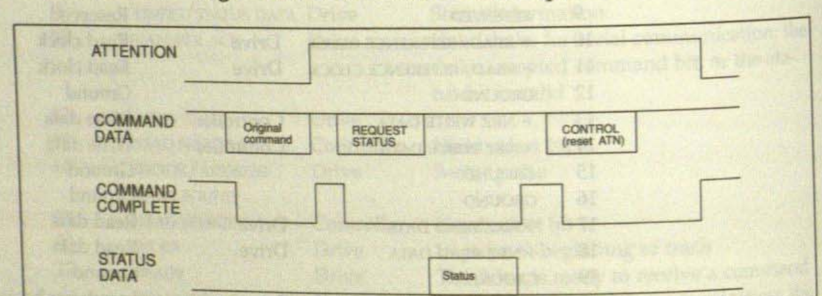


Figure 2.12 ESDI error handling.

The ESDI drive model

For the most part, the ESDI drive model corresponds to that of the ST506 drive. The interface presents to the controller a drive divided into cylinders, tracks, and sectors. An ESDI drive can have 16 groups, each with 16 heads. The implementation of groups is optional. Typically, a maximum of 4095 cylinders are supported. This number can be increased to 65 535 using extended addressing. Up to 256 sectors per track are supported.

Another part of the ESDI model is the defect list format. A cylinder is set aside for the purpose of storing the defect list. This logical cylinder has the number 4095 (65 535), even when there are much fewer cylinders than this. Where this cylinder is located physically on the drive is up to the drive manufacturer.

Commands

The basic format of an ESDI command is shown in Table 2.6. It consists of a 4 bit opcode along with either command extensions or parameter data. There are also commands that can optionally use 16 bit long parameter information. An example of this is the SEEK command. The extended field enables the drive to address more than 4095 cylinders. In order to use this option, the command SET HIGH ORDER VALUE is first sent along with the 4 most significant bits of the parameter, then the actual command follows with the other 12 bits of parameter information (Table 2.7).

Table 2.6 ESDI command format.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	P
Opcode				Modifier				0	0	0	0	0	0	0	0	0
Opcode				Modifier				Extra								
Opcode				Parameter												

Table 2.7 ESDI commands.

Opcode	*Command*	*Optional*	*Modifier*	*Extra*	*Parameter*	*Status*
0000	SEEK				Yes	
0001	RECALIBRATE					
0010	REQUEST STATUS		Yes	Yes		Yes
0011	REQUEST CONFIGURATION		Yes	Yes		Yes
0100	SELECT HEAD GROUP	Yes			Yes	
0101	CONTROL		Yes			
0110	DATA STROBE OFFSET	Yes	Yes			
0111	TRACK OFFSET	Yes	Yes			
1000	INITIATE DIAGNOSTICS	Yes			Yes	
1001	SET UNFORMATTED BYTES/SECTOR	Yes			Yes	
1010	SET HIGH ORDER VALUE	Yes			Bits 0–3	
1011	RESERVED					
1100	RESERVED					
1101	RESERVED					
1110	SET CONFIGURATION	Yes	Yes	Yes		
1111	RESERVED					

Defect list format

A medium defect is a small region on the medium where information cannot be stored reliably. On disk drives, these are spots where the thin layer of magnetic material has been damaged. Since it is not economical to manufacture completely defect-free media, a certain number of defects are tolerated. It then becomes the responsibility of the computer system to deal with them. These locations are identified so that when the drive is formatted sectors can avoid these positions. It is also possible for the computer to find these defects for itself by writing and reading data patterns to the drive. However, the methods used by the manufacturer are much more accurate. They employ analog test equipment to examine the surface of the disk. In fact, this method is able to identify locations that can be read and written to successfully at the moment, but over time will probably lead to data loss.

Defect lists have existed since the inception of disk drives. Originally, they were delivered with the drive in the form of a printed list. There are various methods for describing the precise position of a defect. A popular approach is the 'bytes after index' format. As the name implies, the position of the defect is described by its distance from the index pulse in bytes.

For an ESDI drive, three copies of the list reside on the drive itself. The first list is found on the last cylinder, the second eight cylinders before this, and

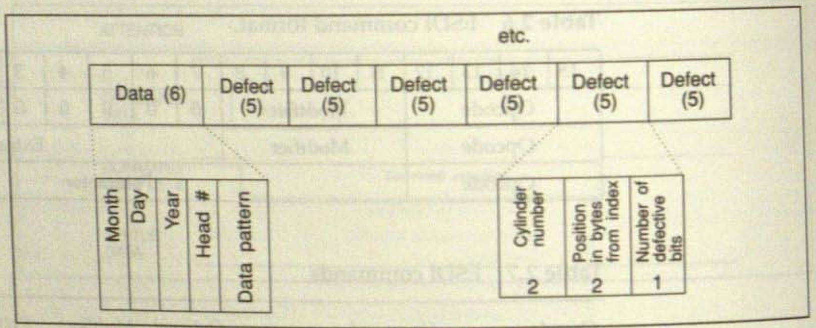


Figure 2.13 ESDI defect list.

the last resides on logical cylinder 4095. This last cylinder is outside the data region and therefore cannot be overwritten. Figure 2.13 shows the format of a sector from the defect list.

Summary The ESDI interface is the most modern of the drive specific interfaces. It may represent the end of development of such a device specific approach. The throughput of up to 3 Mbytes per second is a tremendous improvement over ST506. While SMD can provide a throughput equal to ESDI, it does so at much higher cost. The addition of a protocol layer and command set is a large step in the direction of modern interfaces like IDE and SCSI.

3 Computer buses

In contrast to the peripheral interfaces discussed so far, a computer bus is designed to connect the various components within the computer. All computers utilize a number of internal buses. These buses transport information between the system components like the nervous system of an organism. The more complex a computer system, the more exotic its buses can become (Figure 3.1).

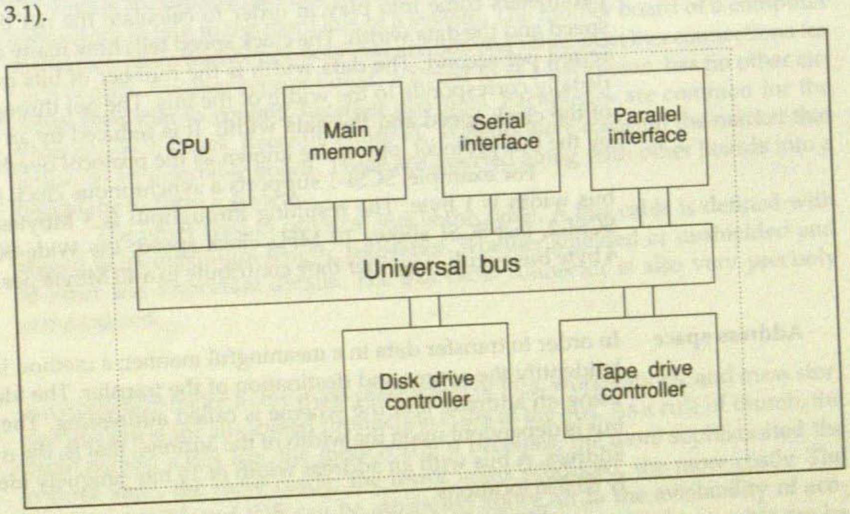


Figure 3.1 Universal bus.

The boundary between a bus and an interface is blurred at best. I consider it an important characteristic of a bus to connect various devices of equal authority. By this measure the IDE interface is excluded, as are all computer memory buses, for that matter. The SCSI bus, on the other hand, clearly matches this definition of a bus. Of course, the discussion of such border cases is purely academic.

The layer model for interfaces can also be applied to computer buses. It is defined by the physical interface, bus protocol and optional device model along with a command set.

A computer bus is built from three basic functional blocks: addressing, data transfer and control. In the literature you will frequently see block diagrams depicting the address, data, and control bus as separate paths. However, since all three of these components depend on the others we will always refer to a computer bus in its entirety.

3.1 Characteristics of buses

There are a number of characteristics that make a bus well suited for a particular application. The most important of these are the throughput, the address space, the real-time performance, the electrical and mechanical characteristics, and the production costs. The following sections examine these in closer detail.

Data throughput Data throughput, also known as bandwidth, is the amount of information the bus can transport per unit of time. It is measured in Mbytes per second. Two parameters come into play in order to calculate the net throughput: the clock speed and the data width. The clock speed tells how many data words are transferred per second. The data width is the number of bits in one data word and usually corresponds to the width of the bus. The net throughput is the product of the clock speed and the data width. It is reduced by an appreciable amount by the bus protocol, otherwise known as the protocol overhead.

For example, SCSI-1 supports a synchronous clock speed of 5 MHz; the bus width is 1 byte. The resulting throughput is 5 Mbytes per second. Under SCSI-2, fast-SCSI allows 10 MHz clock speed; the Wide-SCSI option allows a 4 byte bus width. Together they contribute to a 40 Mbyte per second throughput.

Address space In order to transfer data in a meaningful manner, a method is needed to uniquely identify the source and destination of the transfer. The identification is made using an address, and the scheme is called addressing. The address space of a bus is dependent upon the width of the address; that is, the number of bits in the address. A bus with an address width of 16 bits uniquely identifies 2^{16} or exactly 65 536 locations.

For example, the Q-22 bus of a PDP-11 has an address width of 22 bits. It can therefore address 4 Mbytes of memory. The ISA bus of IBM PC compatibles has 24 address bits and is able to address 16 Mbytes. Modern systems with 32 bit data buses also have 32 bit address buses, corresponding to an address space of 4 Gbytes.

Real-time capabilities Real-time systems are distinguished from other systems by their ability to react to an external event within a given amount of time. This external event may occur at any time. In addition the system may not be able to anticipate the exact moment. A real-time system does not necessarily have to be very fast; however,

Electrical characteristics

Mechanical characteristics

Production cos

its reaction time must be predictable and, of course, adequate for the application. This predictability usually means that a mechanism has been implemented to interrupt running processes. A real-time capable I/O bus must allow, for example, interruption of a lengthy data transfer from disk to tape for an event with higher priority. A bus without this capability could also be used for real-time applications, but only when used for a single device.

Electrical characteristics Two important attributes result directly from the electrical characteristics of a bus: the maximum length of the bus and the integrity of the data. While the bus inside a PC is only a few inches long, I/O buses more than 30 feet in length are often used to connect computers and peripherals. When many such cables are in close proximity to each other, as is often the case, data integrity is a major issue. A bus in a cable duct needs to be less sensitive to electromagnetic interference than a bus that resides inside a metal enclosure.

Mechanical characteristics There are two basic ways to implement a bus physically. Internally, the individual signals are usually part of a printed circuit board. Insertable boards use edge connectors to link to the main bus of a system. The mother board of a computer sometimes has a number of slots that are nothing other than bus connections for such boards. Another type of board, referred to as a backplane, has no other circuitry than that to connect together bus slots. Backplanes are common for the VME and ECB buses. Recently entire PC systems have come on the market that reside on an insertable board. These are inserted along with other boards into a backplane to form a system.

The other type of physical bus is the cable. A bus cable is defined with regard to its maximum length, resistance, whether shielded or unshielded and to other less important details. The bus cable connector is also very precisely standardized.

Production costs An important factor in the mass production of PCs, workstations, and mass storage devices is the associated production costs of the bus. As a rule of thumb, the more signals a bus has, the more costly it becomes; the more sophisticated the control logic, the more costly; the fewer items produced, the more costly. The success of SCSI and IDE can be attributed above all to the availability of economical bus interface components and the fact that a simple ribbon cable can be used to interconnect devices. Moreover, peripheral manufacturers need to equip devices with at most two connector types. Cost is also the reason why SCSI and IDE can coexist in the marketplace: IDE costs slightly less to manufacture than SCSI. In fact, this is often reflected in the price of the IDE and SCSI versions of a particular drive model.

3.2 Specialized buses

The ideal bus, then, would have a large address space, a maximal throughput, and excellent real-time capabilities. There would be no constraints on its length and it would be simple and inexpensive to produce. Unfortunately, such a bus is not even theoretically possible, as the following example shows. A real-time system is characterized by its reaction time to a particular event. This time is independent of, among other things, the length of system buses. Since electrical signals travel with finite speed, as the length of a bus increases so does the reaction time to any signal on the bus. Therefore, it is impossible to design a bus of unconstrained length, which at the same time guarantees an arbitrary reaction time.

For this reason a wide range of buses with differing characteristics have come into existence, each for a particular application.

Memory bus

A memory bus connects the CPU or memory controller to memory. The main requirement of this bus is high bandwidth since every CPU instruction and all data must travel over this path. To meet this constraint, most memory buses are very short. The address space of a memory bus is the physical address space of the computer system.

The CPU of a MicroVAX, for example, has an address width of 32 bits. While this corresponds to an address space of 4 Gbytes, the system physically accommodates only 16 Mbytes. Consequently, the memory bus could be implemented using only 24 address lines.

A memory bus need not implement any real-time or interrupt capabilities. The division of labor is well defined among system components: the CPU makes a request, the memory reads or writes the information. By my definition of a bus at the beginning of this chapter, the memory bus is not a bus at all since in this case the devices do not have equal authority over one another.

I/O bus

An I/O bus connects the CPU with the I/O devices. Here the requirements are somewhat different. The I/O bus must be able to support a variety of devices. It must be able to handle slow as well as fast devices. In addition, there must be a method for determining which device may use the bus when more than one requests use of it. This mechanism is called arbitration. Depending on the application, an I/O bus must also be capable of near real-time performance. This can be extremely important in the area of computer controlled systems. One need only look at the example of a nuclear reactor: it is imperative that the CPU be informed the moment some particular event occurs. All other I/O processes must be suspendable. An I/O bus that allows this must employ interrupt and event priority mechanisms.

Universal bus

Many less sophisticated computer systems use a universal bus to link together the CPU, memory and I/O devices. The goal here is to find the best compromise between bandwidth, real-time capability, and production cost. Examples of universal buses include the ECB bus, VME bus and the ISA bus of IBM AT compatibles. The older PDP 11/73 with its Q-22 bus is another example. In this light, Figure 1.1 can be viewed as a simplified block diagram of an IBM AT. Figure 3.2 shows the structure of a more complex system, the VAX 8800, with several specialized buses.

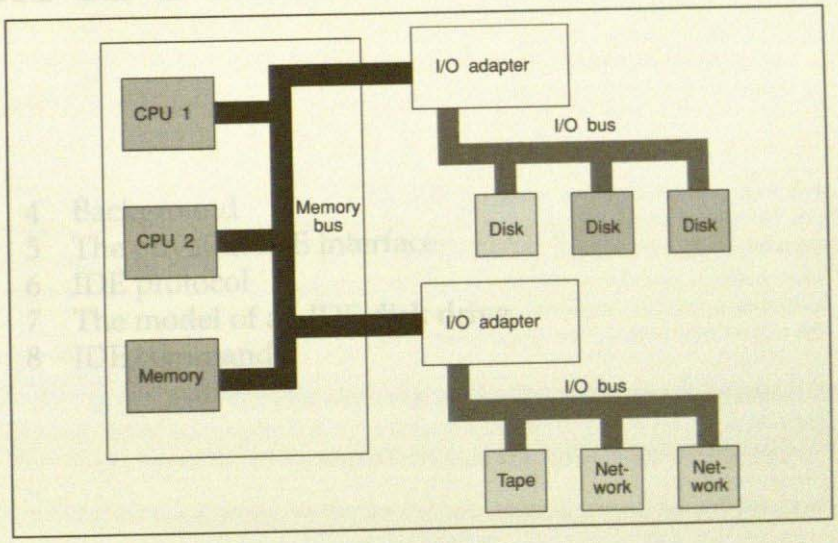
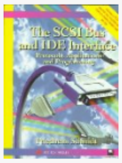


Figure 3.2 Computer system with multiple buses.



The SCSI bus and IDE interface

protocols, applications and programming

Authors/Contributors: Friedhelm Schmidt ; transl. by J. Michael Schultz
Place, publisher, year: WoKingham : Addison-Wesley, 1995
Physical Description: IX, 301 p ; ill + ; 24 cm ; 1 floppy disk ; 3 1/2"
Format: Book

Other Editions

Authors/Contributors
Schmidt, Friedhelm
Schultz, J. Michael

Locations & request | Description | Similar Items | **Fields**

LEADERcam a22 u 4500
001139081496
003CHVBK
00520140831083549.0
008951107s1995 xxk 00 eng|d
010 |**a** 94023813
020 |**a** 0-201-42284-0
035 |**a** (OCoLC)845069869
035 |**a** (NEBIS)001501062
040 |**a** STS-TI |**b** ger |**c** ETH-BIB |**e** ETHICS-ISBD
245 04 |**a** The SCSI bus and IDE interface |**b** protocols, applications and programming |**c** Friedhelm Schmidt ; transl. by J. Michael Schultz
260 |**a** WoKingham |**b** Addison-Wesley |**c** 1995
300 |**a** IX, 301 p |**c** 24 cm |**b** ill + |**e** 1 floppy disk ; 3 1/2"
509 0 |**a** SCSI-Bus und IDE-Schnittstelle
7001 |**a** Schmidt |**D** Friedhelm
7001 |**a** Schultz |**D** J. Michael
950 |**B** NEBIS |**P** 700 |**E** 1- |**a** Schmidt |**D** Friedhelm
950 |**B** NEBIS |**P** 700 |**E** 1- |**a** Schultz |**D** J. Michael
986 |**a** SWISSBIB |**b** 139081496
898 |**a** BK020000 |**b** XK020000 |**c** XK020000
949 |**B** NEBIS |**F** E34 |**j** 004.62 SCHM | 2147
949 |**B** NEBIS |**F** E63 |**j** 681.326.7*1 SCH