IN THE UNITED STATES DISTRICT COURT

FOR THE DISTRICT OF COLUMBIA

IN RE PAPST LICENSING GMBH & CO. KG LITIGATION _____ This Document Relates To: ALL CASES	Misc. Action No. 07-493 (RMC) MDL Docket No. 1880

CAMERA MANUFACTURERS' OPENING MARKMAN BRIEF

TABLE OF CONTENTS

Page

I. PRELIMINARY STATEMENT 1

II. THE LEGAL PRINCIPLES THAT GOVERN *MARKMAN* PROCEEDINGS.....4

A. The Applicable Legal Standards.....4

B. Papst Has Ignored and/or Misapplied the Applicable Legal Standards 5

1. Papst makes improper use of extrinsic evidence. 6

2. Papst's claim construction analysis is improperly directed to the accused Products.....6

3. Papst ignores disclaimers made to the PTO 6

C. Papst Also Errs in Arguing That The Preamble Is Not A Claim Limitation 7

III. THE COURT SHOULD ADOPT THE CAMERA MANUFACTURERS' PROPOSED CONSTRUCTIONS.....9

A. "*host device*" 9

B. "*data transmit/receive device*" 10

C. "*interface device*" 11

D. "*communication between*" [the host device and data transmit/receive device].....14

E. "*first connecting device for interfacing the host device with the interface device*" 15

F. "*second connecting device for interfacing the interface device with the data transmit device*" 17

G. "*first command interpreter*" 18

H. "*second command interpreter*".....20

I. "*multi-purpose interface*" 21

J. "*interfacing*" 22

K. "*wherein the interface device is configured by the processor and*

memory to include a first command interpreter and a second command interpreter” 22

L. “*sends a signal, regardless of the type of the data transmit/receive device attached to the second connecting device of the interface device, to the host device which signals to the host device that it is an input/output device customary in a host device*” 23

M. “*the driver*” 24

N. “*the driver for the input/output [storage] device customary in a host device*” 26

O. “*an input/output [storage] device customary in a host device*” 28

P. “*specific driver for the multi-purpose interface*” 29

Q. *“whereupon the host device communicates with the interface device by means of the driver for the input/output [storage] device customary in a host device”* 30

R. “*the digital data*” 31

S. “*inquiry*” [or “*inquiring*”] 33

T. ‘399 Patent, Claim 2 35

U. “a buffer to buffer data to be transferred between the data transmit/receive device and the host device” or “a data buffer for permitting independence in terms of time of the data transmit/receive device attachable to the second connecting device from the host” 36

V. “*virtual files*” 38

W. “*simulating a virtual file system to the host device*” 40

X. “*the usual driver for the input/output [storage] device*” 42

Y. Miscellaneous Terms that Require No Construction 43

IV. CONCLUSION 43

TABLE OF AUTHORITIES

Page No.

CASES

Accord Exigent Tech., Inc. v. Atrana Solutions, Inc.,
442 F.3d 1301 (Fed. Cir. 2006)..... 6

Accumed LLC v. Stryker Corp.,
483 F.3d 800 (Fed. Cir. 2007)..... 5

Ampex Corp. v. Eastman Kodak Co.,
460 F. Supp. 2d 563 (D. Del. 2006), *aff'd*, 2008 WL 343253 (Fed. Cir. 2008) 32

Bell & Howell Document Mgmt Prods. Co. v. Altek Sys.,
132 F.3d 701 (Fed. Cir. 1997)..... 5, 6, 26, 34

Bell Atlantic Network Serv. v. Covad Comm. Group, Inc.,
262 F.3d 1258 (Fed. Cir. 2001)..... 5

Bicon, Inc. v. Straumann Co.,
441 F.3d 945 (Fed. Cir. 2006)..... 8

Boss Control, Inc. v. Bombardier, Inc.,
410 F.3d 1372 (Fed. Cir. 2005)..... 5

Catalina Marketing Int'l v. Coolsavings.com, Inc.,
289 F.3d 801 (C.A. Fed. Ill. 2002)..... 9

Chef America, Inc. v. Lamb-Weston, Inc.,
358 F.3d 1371 (Fed. Cir. 2004)..... 35, 41

Computer Docking Station Corp. v. Dell, Inc.,
519 F.3d 1366 (Fed. Cir. 2008)..... 9

CSS Fitness, Inc. v. Brunswick Corp.,
288 F.3d 1359 (Fed. Cir. 2002)..... 4

Eaton Corp. v. Rockwell Int'l Corp.,
323 F.3d 1332 (Fed. Cir. 2003)..... 8

Elekta Instrument S.A. v. O.U.R. Scientific Int'l, Inc.,
214 F.3d 1302 (Fed. Cir. 2000)..... 35

Golight, Inc. v. Wal-Mart Stores, Inc.,
355 F.3d 1327 (Fed Cir. 2004)..... 4

Halliburton Energy Serv., Inc. v. M-I LLC,
514 F.3d 1244 (Fed. Cir. 2008)..... 42

Innova/Pure Water, Inc. v. Safari Water Filtration Sys., Inc.,
381 F.3d 1111 (Fed. Cir. 2004)..... 4

Inpro II Licensing S.A.R.L. v. T-Mobile USA, Inc.,
450 F.3d 1350 (Fed. Cir. 2006)..... 4

Kopykake Enterprises, Inc. v. Lucks Co.,
264 F.3d 1377 (Fed. Cir. 2001)..... 27

Kropa v. Robie,
187 F.2d 150 (CCPA 1951) 7

Lemelson v. General Mills Inc.,
968 F.2d 1202, 1207-08 (Fed. Cir. 1992) 33

Mangosoft v. Oracle Corporation,
525 F.3d 1327 (C.A. Fed. Cir. 2008) 16, 18

Markman v. Westview Instruments, Inc.,
52 F.3d 967 (Fed. Cir. 1995) (en banc), aff'd, 517 U.S. 370 (1996) 1, 2, 3, 4, 5, 6

Merck & Co. v. Teva Pharm. USA, Inc.,
395 F.3d 1364 (Fed. Cir. 2005)..... 35

Microsoft Corp. v. Multi-Tech Sys. Inc.,
357 F.3d 1340 (Fed. Cir. 2004)..... 1, 4

Muniauction, Inc. v. Thomson Corp.,
__ F.3d __, 2008 WL 2717689 (Fed. Cir. July 14, 2008)..... 27

Mycogen Plant Science v. Monsanto Co.,
243 F.3d 1316 (Fed. Cir. 2001)..... 31

Omega Eng'g Inc. v. Raytek Corp.,
334 F.3d 1314 (Fed. Cir. 2003)..... 5

PC Connector Solutions LLC v. SmartDisk Corp.,
406 F.3d 1359 (Fed. Cir. 2005)..... 26, 27

Phillips v. AWH Corp.,
415 F.3d 1303 (Fed. Cir. 2005)..... 4, 5, 12, 26, 34, 36

Pitney Bowes, Inc. v. Hewlett-Packard, Co.,
182 F.3d 1298 (Fed. Cir. 1999)..... 7

Process Control Corp. v. HydReclaim Corp.,

190 F.3d 1350 (Fed. Cir. 1999)..... 32

Renishaw PLC v. Marposs Societa' Per Azioni,
158 F.3d 1243 (Fed. Cir. 1998)..... 43

Research Plastics, Inc. v. Federal Packaging Corp.,
421 F.3d 1290 (Fed. Cir. 2005)..... 7

Solomon Technologies, Inc. v. Int'l Trade Commission,
524 F.3d 1310 (Fed. Cir. 2008)..... 19

SRI Int'l v. Matsushita Elec. Corp. of Am.,
775 F.2d 1107 (Fed. Cir. 1985)..... 6, 12

Stumbo v. Eastman Outdoors, Inc.,
508 F.3d 1358 (Fed. Cir. 2007)..... 28

Verizon Services Corp. v. Vonage Holdings Corp.,
503 F.3d 1295 (Fed. Cir. 2007)..... 20

Verizon Servs. Corp. v. Vonage Holdings Corp.,
503 F.3d 1295 (Fed. Cir. 2007)..... 12

Vitronics Corp. v. Conceptronic, Inc.,
90 F.3d 1576 (Fed. Cir. 1996)..... 4, 5, 6

STATUTES AND RULES

35 U.S.C. § 112..... 36, 40, 42

OTHER AUTHORITIES

Amer. Heritage Dict. of Computer and Internet Words 59 (2001)..... 16

Amer. Heritage Dict. of Computer Words 54 (1995)..... 16

American National Standard for Information Systems – Small Computer System Interface – 2 at 96-100 (1994)..... 34

IEEE 100 The Authoritative Dictionary of IEEE Standards Terms (7th ed.) (2000)..... 25

MANUAL OF PATENT EXAMINING PROCEDURE § 2173.05(e)..... 32

Michael Tasler, *Design and Construction of a Universal Data Acquisition and Control System for Scanning Probe Microscopy* (1995)..... 37

OXFORD DICTIONARY OF COMPUTING (4th ed. 1996)..... 37

Random House Webster's Computer & Internet Dictionary (3d ed.) 1999 39

S.M.H. Collin, Dictionary of Personal Computing and The Internet (1997)..... 39

THE COMPUTER GLOSSARY 38 (1998) 17

The New IEEE Dictionary of Electrical & Electronics Terms (8th ed.) 1998 39

The New IEEE Standard Dictionary
of Electrical & Electronics Terms (5th ed.) (1993) 25

William E. Ham, "Recent Advances in Basic Physical Technology for
Parallel SCSI," Digital Technical Journal 9:3, p. 18 (1997)..... 17

I. PRELIMINARY STATEMENT

Papst devotes the first seven pages of its *Markman* Brief to a story about how the named inventor of the patents-in-suit, Michael Tasler, purportedly solved a "Specific Software Driver Problem" "faced by everyday computer users" in the late 1990s, and about how the digital cameras it now accuses of infringement use a "strategy" similar to that employed by Tasler. Even if it were true, this story would be of little or no relevance to the task of claim construction now at hand. Governing precedent makes clear that the Court's focus in the *Markman* process must be on the *intrinsic record* of the patents-in-suit, not on a story unsupported by that record or on the accused products. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1315-17 (Fed. Cir. 2005) (en banc) ("The descriptive part of the specification aids in ascertaining the scope and meaning of the claims inasmuch as the words of the claims must be based on the description. The specification is, thus, the primary basis for construing the claims."). But in its *Markman* brief, Papst departs from this basic tenet. Moreover, the tale told in its brief is also demonstrably wrong in a number of important respects.

First, the computer industry, led by Intel, Microsoft and others was well on the way to implementing a permanent solution to the "problem" in question years before Tasler filed his original patent application. One need look no further than the prosecution file of the patents-in-suit to see that the "Specific Software Driver Problem" *had already been recognized and solved* by others well before Tasler supposedly recognized and solved it in exactly the same way. *See, e.g.*, Ex. B, U.S. Patent No. 5,506,692 (cited prior art to the '399 patent) at 1:32-51 (recognizing the specific driver problem) and 4:20-23: (proposing the same solution: "The image scanner 20 emulates the file system of the 'UNIX' *as if it were a hard disc*. Accordingly, the image scanner 20 looks like the hard disc from the workstation 21 and can be handled as the hard disc."). The prosecution file reveals that Tasler merely adapted these published solutions for the specific

applications he had in mind (none of which involved digital cameras).

Second, the "interface device" patented by Mr. Tasler is a stand-alone device that acts as an intermediary between 1) a computer and 2) a variety of different types of "data transmit/receive devices." '399, col. 1:34-54. As examples of "data transmit/receive devices" that can be connected to a computer via the claimed "interface device," the Tasler patents disclose "a diagnostic radiology system in a medical engineering environment" and a "multimeter." *Id.* Contrary to the story told in Papst's *Markman* Brief, the interface device disclosed by Papst had no relation to "every day computer users." The problems discussed and the solutions offered in the asserted patents are in the context of simplifying the data acquisition work of "field service technicians." '399, col. 1:39-46, 60-64. There is no mention of digital cameras anywhere in any of the Tasler patents and no suggestion that the "interface device" patented by Mr. Tasler might be permanently affixed to a single "data transmit/receive device."

Third, in contrast with the description provided by Papst, the intrinsic record of the patents-in-suit repeatedly describes Mr. Tasler's "interface device" as "flexible" and "universal" with respect to the types of data transmit/receive devices that could be attached to it, thereby freeing technicians of the need to load special purpose drivers on the computer to enable communication with any given data transmit/receive device. Thus, while it adopted and used the "specific driver" solution developed by others, the "interface device" disclosed in the Tasler patents has a different and more specific purpose. As exemplified by the title of both patents-in-suit ("Flexible Interface for Communication Between a Host and an Analog I/O device Connected to the Interface Regardless of the Type of I/O Device"), and as explained throughout the common specification of the '399 and '449 patents ("specification"), the Tasler "interface device" is designed to be readily attached and detached from different types of

"transmit/receive" devices. *See, e.g.*, '399 patent, col. 1:34-35, 1:55-64 ("[A]n interface may be put to totally different uses. It is therefore desirable that an interface be sufficiently flexible to permit attachment of very different electrical or electronic systems to a host device by means of the interface.") The language of the claims and the title of the patents-in-suit reflect Mr. Tasler's objective of making his interface device "universal" so that it could connect different types of data transmit/receive devices to a computer connected via the patented "interface device."

In addition to mischaracterizing the nature of Tasler's alleged invention, Papst also misapplies the applicable principles of claim construction. Although Papst's Brief acknowledges at the outset the claim construction principles that govern this Court's *Markman* proceedings, the "facts" and "arguments" presented by Papst do not adhere to those principles. Papst, for example, relies on the declaration of a proffered expert to support proposed constructions that are at odds with the intrinsic evidence, including the claim language and the specification.[1] Papst also errs by repeatedly seeking to support and tailor its proposed constructions by reference to the products it accuses of infringement, contrary to established Federal Circuit precedent. In contrast, the Camera Manufacturers have proposed claim constructions that are consistent with, and indeed compelled by, the intrinsic evidence. For the reasons set forth below, the Court should adopt the Camera Manufacturers' proposed constructions.[2]

---

[1] Although the Camera Manufacturers do not believe expert testimony is needed or appropriate at this stage to resolve disputed issues, they may use an expert to assist with the neutral tutorial that the parties will present to the Court on September 3, 2008. The Camera Manufacturers reserve the right to present expert testimony, in the event the Court decides to permit such testimony at the *Markman* hearing.

[2] Exhibit A hereto is an Appendix of the Camera Manufacturers' Proposed Constructions.

II. THE LEGAL PRINCIPLES THAT GOVERN *MARKMAN* PROCEEDINGS

A. The Applicable Legal Standards

Claim construction is a matter of law to be decided by the Court. *See, e.g., Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 979 (Fed. Cir. 1995) (en banc), *aff'd*, 517 U.S. 370 (1996). It is further well-settled that a court should construe a disputed claim term, whenever possible, by considering the intrinsic evidence, which comprises the claims themselves, the specification, and the prosecution history. *E.g., Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1582 (Fed. Cir. 1996).

First, the Court must look to the language of the claims. "It is a 'bedrock principle' of patent law that 'the claims of a patent define the invention to which the patentee is entitled the right to exclude.'" *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312 (Fed. Cir. 2005) (en banc) (*quoting Innova/Pure Water, Inc. v. Safari Water Filtration Sys., Inc.*, 381 F.3d 1111, 1115 (Fed. Cir. 2004)). There is a presumption that words in a claim carry their plain and ordinary meaning and should be construed as they would be understood by a person skilled in the relevant art at the time of the invention, i.e., as of the effective filing date of the patent application. *Golight, Inc. v. Wal-Mart Stores, Inc.*, 355 F.3d 1327, 1332 (Fed Cir. 2004); *CSS Fitness, Inc. v. Brunswick Corp.*, 288 F.3d 1359, 1366 (Fed. Cir. 2002); *Phillips*, 415 F.3d at 1313..

Next, the specification is the "single best guide to the meaning of a disputed term," and "[u]sually it is dispositive." *Phillips*, 415 F.3d at 1315. Although a patent's scope is not necessarily limited to an embodiment discussed in the specification, "neither do the claims enlarge what is patented beyond what the inventor has described as his invention." *Inpro II Licensing S.A.R.L. v. T-Mobile USA, Inc.*, 450 F.3d 1350, 1355 (Fed. Cir. 2006). Accordingly, the claims cannot be construed as broader than what is described in the specification as the invention. *Microsoft Corp. v. Multi-Tech Sys. Inc.*, 357 F.3d 1340, 1347 (Fed. Cir. 2004).

Finally, the Court must also review the patent's prosecution history (or file history), which is "the complete record of all proceedings before the Patent and Trademark Office ("PTO"), including any express representations made by the applicant regarding the scope of the claims." *E.g., Vitronics*, 90 F.3d at 1582; Papst Br. at 10-11. A patent applicant can limit claims during prosecution by, for example, altering claim language to overcome an examiner rejection, arguing to overcome or distinguish a reference, or disavowing claim coverage. *Omega Eng'g Inc. v. Raytek Corp.*, 334 F.3d 1314, 1323-25 (Fed. Cir. 2003).

If the meaning of the claim can be discerned from the intrinsic evidence, then resort to extrinsic evidence is unnecessary and improper. *Boss Control, Inc. v. Bombardier, Inc.*, 410 F.3d 1372, 1377 (Fed. Cir. 2005). *See also Bell Atlantic Network Serv. v. Covad Comm. Group, Inc.*, 262 F.3d 1258, 1269 (Fed. Cir. 2001); *Bell & Howell Document Mgmt Prods. Co. v. Altek Sys.*, 132 F.3d 701, 706 (Fed. Cir. 1997). In such circumstances, dictionaries or comparable sources may be "useful to assist in determining the commonly understood meaning of words and have been used both by [the Federal Circuit] and the Supreme Court in claim interpretation." *Phillips*, 415 F.3d at 1322; s*ee also Accumed LLC v. Stryker Corp.*, 483 F.3d 800, 809 n.2 (Fed. Cir. 2007). "A dictionary definition has the value of being an unbiased source 'accessible to the public in advance of litigation.'" *Phillips*, F.3d at 1322. As the Camera Manufacturers have explained in their briefs regarding *Markman* expert testimony, the Federal Circuit has warned courts to be cautious in considering expert statements during *Markman* proceedings, particularly "conclusory, unsupported assertions by experts as to the definition of a claim term." *Id*.

B. Papst Has Ignored and/or Misapplied the Applicable Legal Standards

Papst's proposed constructions suffer from a number of defects, including that Papst repeatedly ignores and/or misapplies the governing legal standards discussed above. Those defects infect Papst's proposed constructions for many of the specific terms discussed in Section

III, *infra*. For example:

1. Papst makes improper use of extrinsic evidence.

Throughout its *Markman* brief, Papst ignores the claim language and the specification and instead repeatedly offers expert testimony to support its proposed constructions. That approach is wrong as a matter of law because intrinsic evidence should be the focus of the claim construction analysis and expert testimony cannot be used to vary the meaning of claim terms provided in the intrinsic record. *Vitronics*, 90 F.3d at 1584-85; *Bell & Howell*, 132 F.3d at 706.

2. Papst's claim construction analysis is improperly directed to the accused Products

As noted above, neither the specification nor the prosecution history contains a single reference to digital cameras. Nonetheless, Papst repeatedly seeks to justify its proposed constructions by comparing the language of the claims to the accused products. That approach is wrong as a matter of law. Claim construction should be undertaken *independent of* any facts or considerations regarding how the claims may (or may not) read on the accused products. *SRI Int'l v. Matsushita Elec. Corp. of Am.*, 775 F.2d 1107, 1118 (Fed. Cir. 1985) ("A claim is construed in the light of the claim language, the other claims, the prior art, the prosecution history, and the specification, *not in light of the accused device*.") (emphasis added). *Accord Exigent Tech., Inc. v. Atrana Solutions, Inc.*, 442 F.3d 1301, 1309 n 10 (Fed. Cir. 2006) ("the district court's grant of summary judgment was in error because the court considered the accused product in rendering its claim construction.").

3. Papst ignores disclaimers made to the PTO

As discussed further below, the patentee presented arguments to the PTO to distinguish the prior art and obtain allowance of the patents-in-suit. In its *Markman* brief, Papst ignores those limiting statements and, in a misguided attempt to support its infringement contentions,

improperly seeks to broaden the meaning of the claims through the claim construction process. *See Research Plastics, Inc. v. Federal Packaging Corp.*, 421 F.3d 1290, 1296 (Fed. Cir. 2005).

C. Papst Also Errs in Arguing That The Preamble Is Not A Claim Limitation

The independent claims of the '399 and '449 patents each disclose an "interface device" for "communication between" a "host device" and a "data transmit/receive device." The preambles not only contain these specific limitations but also further define the type of host device and data transmit/receive device required to practice the claims. "If the claim preamble, when read in the context of the entire claim, recites limitations of the claim, or, if the claim preamble is 'necessary to give life, meaning, and vitality' to the claim, then the claim preamble should be construed as if in the balance of the claim." *Pitney Bowes, Inc. v. Hewlett-Packard, Co.*, 182 F.3d 1298, 1305 (Fed. Cir. 1999) (*quoting Kropa v. Robie*, 187 F.2d 150, 152 (CCPA 1951)).

The preambles for each of the asserted claims describe structures that comprise the claimed invention as well as the relationships among those structures, as illustrated by the following preamble to '399 patent, claim 1:

> *An interface device* for communication between
>
> [A] *a host device*, which comprises
> [1] *drivers for input/output devices* customary in a host device and
> [2] *a multipurpose interface*,
>
> and [B] *a data transmit/receive device*, the data transmit/receive device being arranged for providing analog data, comprising . . .

Each of the above italicized terms describes structure (as opposed to "merely the intended purpose") of the claimed invention. Moreover, the body of '399 patent, claim 1, refers back repeatedly to these structures and incorporates them by reference, by using them with a definite article ("the"). *See, e.g.*, '399 patent, claim 1 ("a first connecting device for interfacing *the host device* with *the interface device* via *the multipurpose interface* of the host device") (emphasis

added). In referring to "*the* host device," *the* interface device," and "*the* multipurpose interface," this particular element of claim 1 incorporates structures described more fully in the preamble, which provides antecedents for those and other terms used in the body of the claims. *See, e.g.*, '399, col. 12:42-13:13 (claim 1); '449, col. 11:46-12:6 (claim 1). Where, as here, preambles provide an antecedent basis for terms found in the body of the claims, they "act as a necessary component of the claimed invention," and thus are claim limitations. *Bicon, Inc. v. Straumann Co.*, 441 F.3d 945, 952-53 (Fed. Cir. 2006) (quoting *Eaton Corp. v. Rockwell Int'l Corp.*, 323 F.3d 1332, 1339 (Fed. Cir. 2003)) (holding a preamble was limiting due to references in the claims that derived their antecedent basis from the preamble).

The preambles of the claims asserted by Papst also give life, meaning and vitality to the claims. The preambles make clear that the "interface device" is for *communicating between* a host device and a data transmit/receive device—i.e., it is neither the host nor the data/transmit receive device, but rather a separate device that enables active communication between the other devices. *See, e.g.*, '399, col. 1:9-13. Indeed, the very point of this invention is to allow different existing "data transmit/receive devices" to be attached to different existing "host devices" via a distinct, new type of "interconnect device." *See id.* at col. 1:9-2:14; *see also* Papst Br. at 2.

Moreover, during prosecution of the patents-in-suit, the patentee emphasized the importance of the preambles to distinguish the prior art. For example, when prosecuting the '399 patent, the applicant added the phrase, "the data transmit/receive device arranged for providing analog data," to the preamble of the independent claims. *See* Ex. C, '399 File History, March 18, 2002 Response at 5, 7-9. He then argued that the McNeill, Jr. et al. reference cited by the Examiner as potentially invalidating was different because it "does not disclose that the data transmit/receive device is arranged for providing analog data." *Id.* at 5. Such statements

confirm that the preambles disclose features of the claimed invention and thus are limiting. *E.g.*, *Computer Docking Station Corp. v. Dell, Inc.*, 519 F.3d 1366, 1375 (Fed. Cir. 2008) (holding that the preamble term "portable computer" is a claim limitation because the patentee's statements in the written description and prosecution "emphasize[d] this feature of the invention"). Here, as in *Computer Docking*, the preambles of the independent claims are a "necessary and defining aspect of the invention," and should therefore be construed as claim limitations. *Id.*

The case law relied upon by Papst is inapposite. *See* Papst Br. at 14-15. In contrast to the applicant here, the applicant in *Catalina Marketing* "did not rely on [the preamble] to define its invention nor is the phrase essential to understand limitations or terms in the claim body," and "applicants also did not rely on the preamble phrase to distinguish over the [prior art] patent." *Catalina Marketing Int'l v. Coolsavings.com, Inc.*, 289 F.3d 801, 810 (Fed. Cir. 2002).

III. THE COURT SHOULD ADOPT THE CAMERA MANUFACTURERS' PROPOSED CONSTRUCTIONS

A. *"host device"*

PROPOSED CONSTRUCTION: A general purpose computer that connects to and controls the operation of peripherals.

The Court should rule that a "host device" is "a general purpose computer that connects to and controls the operation of peripherals." Papst agrees that the "host device" is a general purpose computer. Papst Br. at 15. The general purpose computer of the patents-in-suit, however, must also connect to and control the operation of peripherals, as other limitations of the claims require the host device to have "customary" drivers for these peripherals. *See, e.g.*, '399, col. 13:5-8; col. 4:27-39 (describing peripherals such as hard disks, graphics devices, printer devices, floppy disk drives, CD-ROM drives and tape drives). Papst's proposed construction fails to clarify this issue, merely providing that "hardware devices" may be attached.

B. *"data transmit/receive device"*

PROPOSED CONSTRUCTION: A device that transmits data to and receives data from the host device when connected to the host device by the interface device.

The asserted claims all require a "data transmit/receive device" that communicates with a "host device" via an "interface device." *See* Section II.C above regarding limiting preambles. The plain language requires a device that can and does "receive" data – it does not call for only a "data transmit device" or a "transmitter," but instead requires the claimed element to be both a transmitter and a receiver. Had the patentee intended to cover devices that could be *only* transmitters or *only* receivers, then he easily could have done so by claiming either a "data transmit device" or a "data receive device." Alternatively, if he wanted to cover a device that could function as either a transmitter or receiver (but not both), he could have used the language such as "data transmit or receive device."

Papst seeks to interpret the slash mark ("/") as synonymous with "or," but that interpretation does not comport with the language of the claims. Indeed, in other claims where the patentee wanted to express the disjunctive, he did so expressly by using the word "or." *See, e.g.*, claim 10 of the '399 patent and claims 2 and 14 of the '449 patent. The specification also describes the "transmit/receive device" as accomplishing a data *interchange, involving transmitting and receiving*, rather than allowing only a one-way flow of data.

> [a]n important advantage of the interface device of the present invention is that it also permits extremely high data transfer rates by using, *for data interchange*, the host device-own BIOS routines…

'399, col. 8:43-46 (emphasis added). If the data transmit/receive device may only transmit, as Papst now argues, then there would be no "interchange" of data but instead just transfer of data from the transmit/receive device to the host. In addition, as discussed more fully below, there would be no "communication between" the data transmit/receive device and the host.

The specification states that:

> [t]he second connecting device can be attached by means of an output line 16 to a *data transmit/receive device which is to receive data from the host device* or from which data is to be read, i.e. acquired, and transferred to the host device. '399, col. 5:56-60; '499, col. 4:55-59 (emphasis added).

Contrary to Papst's proposed interpretation, Papst Br. at 15, this part of the specification shows that the user may attach the data transmit/receive device and do *either* of two things: transmit data to the host from the transmit/receive device, or receive data in the transmit/receive device from the host. Moreover, the specification distinguishes between "data" and "commands." *See, e.g.*, '399, col. 6:55-58; col. 11:58-65; col. 1:44-54; col. 2:33-36; col. 7:1-22; col. 11:19-25; and col. 11:32-34. The intrinsic evidence thus teaches that the *data* transmit/receive device transmits *data* to and receives *data* from the host device.

C. *"interface device"*

PROPOSED CONSTRUCTION: A stand-alone device that a user can readily physically connect to and disconnect from a host device and a data transmit/receive device and that directs communication between these devices when they are connected.

Papst admits that the interface device "in the context of [the] patents-in-suit, is *a hardware device* that serves as a *bridge* between a computer … and a data device that acquires or transmits data." Papst Br. at 2 (emphasis added). It further acknowledges that the host device is a stand-alone system, such as a personal computer. Papst Br. at 15, 26. However, Papst then eliminates any meaningful distinction between the alleged "interface device" and the alleged "data device" by arguing that the interface device need not be stand-alone, but can be a part of another device. Papst Br. at 6.

The problem with Papst's argument is that instead of pointing to any teachings found in the patents that would vary the ordinary meaning of this term in the fashion Papst suggests, Papst improperly points to the accused products. "A claim is construed in light of the claim language,

the other claims, the prior art, the prosecution history, and the specification, *not in light of the accused device*." *SRI Int'l*, 775 F.2d at 1118 (emphasis added).

The "interface device" limitation should be read in light of the patent specification, not the accused products. *See, e.g., Phillips*, 415 F.3d at 1313-14. Not only does the specification always describe the interface device as a stand-alone device (*see, e.g.,* '399 patent, Title, Abstract, col. 1:1-14, 3:22-5:32, 5:47-63, Figs. 1-2 and accompanying text), but it also touts the advantages and "enormous" benefit of the interface device being a stand-along device:

> In the interface device according to the present invention an enormous advantage is to be gained, as apparent in the embodiment described in the following, in separating the actual hardware required to attach the interface device 10 to the data transmit/receive device from the communication unit … as this allows a plurality of dissimilar device types to be operated in parallel in identical manner. '399, col. 8:23-31.
>
> The interface device 10 thus provides a universal solution which can cover the entire spectrum of possible data transmit/receive devices. '399, col. 12:37-41.

Indeed, the stand-alone nature of the interface device is what solves the problems identified in the prior art: "It is therefore desirable that an interface be sufficiently flexible to permit attachment of very different electrical or electronic systems to a host device by means of the interface," such that a "universal method of operating the interface be provided for a large number of applications." *Id.* at col. 1:56-64; col. 12:23-25. Such statements from the specification explain that the interface device is a stand-alone device, so that it can be connected to multiple other dissimilar devices as needed.

Moreover, affirmative statements in the specification about "the present invention" control the scope of the claims on claim construction. "When a patent thus describes the features of the 'present invention' as a whole, this description limits the scope of the invention." *Verizon Servs. Corp. v. Vonage Holdings Corp.*, 503 F.3d 1295, 1308 (Fed. Cir. 2007) (reversing claim construction that failed to require limitations described by specification as "the present

invention"). In this case, the patentee claimed that an enormous advantage of the "present invention" was to "allow[] a plurality of dissimilar device types to be operated in parallel." Papst's proposed construction cannot be accepted because it seeks to eliminate this "enormous" advantage and improperly read the claims as covering an "interface device" that is an integral piece of the data transmit/receive device, rather than a separate device.

The prosecution history of the '399 patent further supports the Camera Manufacturers' proposed construction. Tasler amended claim 1 to recite "wherein the first command interpreter is configured in such a way that the command interpreter, when receiving an inquiry from the host device as to [the] a type of a device attached to the multi-purpose interface of the host device." Ex. C, '399 File History, March 18, 2002 Response at 7. In the same response, Tasler stated "it is clear that the data transmit/receive device *to be connected to* the second connecting device of the subject interface provides analog data." Ex. C, '399 File History, March 18, 2002 Response at 5 (emphasis added). Together, this claim amendment – from *the* type of device to *a* type of device – and the statement that the data transmit/receive device is *to be connected to* the interface, not that it *is connected to* the interface, show that Tasler believed his "interface device" was not permanently affixed to a single data/transmit receive device. Rather, Tasler's interface device is capable of being readily *connected to* different types of data transmit/receive devices.

This construction is also supported by the plain language of the claims, which separately recite a host device, a data transmit/receive device and an interface device as distinct devices connected by the first and second connecting devices of the interface device. In addition, the claims require that the interface device operates in a particular manner "*regardless of the type of the data transmit/receive device attached to the second connecting device*." This claim language defining the operation of the interface device makes no sense if the interface device is

permanently connected to a single data transmit/receive device. But, the language is perfectly understandable if the interface device is capable of being readily *attached to* different types of data transmit/receive devices, just as Tasler indicated it was during the prosecution of the '399 patent.

D. *"communication between"* [the host device and data transmit/receive device]

PROPOSED CONSTRUCTION: for transmitting of information bidirectionally and actively between two devices

The communication must be bidirectional. The plain meaning of the claim term "between" requires a two-way, bidirectional flow of data. Had the patentee meant there to be only one-way data flow, he would have used the words "to," "from," or even "to or from" rather than "between." Indeed, the patentee used the words "from" and "to" later in the claims to show a contrast with the "between" language used for the claimed "communication." *See, e.g.*, "a data request command *from* the host device to the type of input/output device…" and "transfer of the digital data *to* the host device." '399, col. 13:10-13 (emphasis added).

The specification also strongly supports bidirectional communication between the host and the data transmit/receive device. For example, the "FIELD OF INVENTION" states:

> [t]he present invention relates to the transfer of data and in particular to interface devices for communication between a computer or host device and a data transmit/receive device *from which data is to be acquired or with which two-way communication is to take place*. '399, col. 1:9-14 (emphasis added.)

Thus, two modes of "communication between" a host and a data transmit/receive are identified in the specification, namely, 1) the host acquiring data from the data transmit/receive device and 2) two-way "communication" between the host and the data transmit/receive device. The patentee used the word "communication" in the claim rather than addressing mere data acquisition, and the specification shows that "communication" to be "*two-way communication*." Moreover, all arrows representing data flow in the figures of the patents-in-suit are *bidirectional*

between the host and the data transmit/receive device. *See, e.g.*, Figs. 1 & 2.

The communication must also be active. The plain meeting of "communication" requires an active engagement. Thus, for two devices to be in communication, there must be some real-time activity between them. By contrast, storage of information by one device, and later acquisition of that information by another device, does not place those two devices in communication under any common understanding of the word "communication." As a simple illustration of this point, if a person in Washington takes data from her computer and stores it in a memory, and then ships the memory to a friend in Beijing, where the friend then uses his computer to access the data, one would not ordinarily consider the two computers to be in "communication" with each other.

This is consistent with the consistent usage of this term in the specification. The specification teaches that communications between the host and the data transmit/receive device must be active to achieve the desired high data transfer rate (one of the claimed objects of the invention):

> The data transmit/receive device itself can also communicate *actively* with the host device via the first and second connecting device, as described in more detail in the following.

'399, col. 5:60-63 (emphasis added). *See also* col. 1:26-30; col. 11:53-65; col. 12:8-22.

E. *"first connecting device for interfacing the host device with the interface device"*

PROPOSED CONSTRUCTION: A physical plug or connector for permitting a user to readily attach and detach the interface device with the host device.

As the claim language explicitly requires, the first connecting device must "interface[] the host device with the interface device via the multi-purpose interface of the host device." Throughout the specification, the patentee uses the term "interface" to mean physically connecting. *See, e.g.,* '399, col. 9:30 – col. 10:14. Indeed, consistent with the fact that the

interface device of the patent is a stand-alone device that is connectable with *any host* device, the specification discloses that the first connecting device is a 50-pin SCSI (small computer system interface) connector. *See* '399, col. 9:30-48 (also describing a 25-pin D-shell connector associated with an EPP (enhanced parallel port), and optionally a further 25-pin connector which permits the attachment of eight digital outputs and eight digital inputs at the host device). A connector is simply a physical plug or socket. *See, e.g.*, Ex. D, AMER. HERITAGE DICT. OF COMPUTER AND INTERNET WORDS 59 (2001) ("A coupler used to join two cables or to plug a cable into a port or interface."); Ex. E, AMER. HERITAGE DICT. OF COMPUTER WORDS 54 (1995) (same). The examples given in the patent are entirely consistent with these dictionary definitions and the manner in which one of ordinary skill in the art would understand the term "connector," i.e., that it requires a physical plug or socket. While the first connecting device can certainly "include" other structures and circuits (such as the SCSI interface), it would be improper not to give meaning to the term "first connecting device" as well. *See, e.g., Mangosoft v. Oracle Corporation*, 525 F.3d 1327, 1330 (Fed. Cir. 2008) (construction that rendered term superfluous was improper).

Papst's proffered interpretation of "first connecting device" is wrong because it cites to the SCSI and EPP circuit devices as examples of a "first connecting device," but does not require that the "first connecting device" be a physical plug or socket. *See* Papst Br. at 16–17. Papst's interpretation should not be adopted because eliminating the physical connector requirement is inconsistent with the claims, the specification, relevant technical dictionaries, and the understanding of one of ordinary skill in the art, as well as the patent's asserted advantage of providing a stand-alone interface device which is connectable with any host device and "whose use is host device - independent." *See* '399, col. 3:24–27.

F. *"second connecting device for interfacing the interface device with the data transmit device"*

PROPOSED CONSTRUCTION: A physical plug or socket for permitting a user to readily attach and detach the interface device with a plurality of dissimilar data transmit/receive devices.

The "*second connecting device*" allows the stand alone interface device to be attached to different data transmit/receive devices, thus providing "a *universal solution which can cover the entire spectrum of possible data transmit/receive devices.*" '399, col. 12:38-41 (emphasis added). *See also* '399, col. 12:53-59. Like the first connecting device, the second connecting device also must be a physical plug or socket.

The patent makes clear that this ordinary meaning of connector was intended. The connectors cited in the patents are a "50-pin SCSI connector" to connect to the host and an "8 BNC inputs" connector to connect to the data transmit/receive device. *See, e.g.,* '399, col. 9:30-34; 9:49-64; Fig. 2. These were common connectors at the time, and are each depicted below:



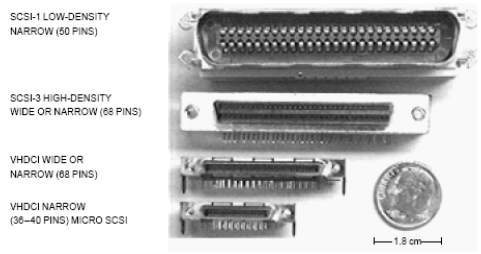


Figure 8
External SCSI Connectors

Ex. D, AMER. HERITAGE DICT. OF COMPUTER AND INTERNET WORDS 32 (2001); Ex. F, William E. Ham, "Recent Advances in Basic Physical Technology for Parallel SCSI," Digital Technical Journal 9:3, p. 18 (1997).

Rather than adopting a construction for "second connecting device," Papst improperly attempts to read the term "second connecting device" entirely out of the claims of the '399 patent, by defining it solely as the rest of the structure recited in this element of claim 1, i.e., an analog-

to-digital converter and sampling circuit. Papst Br. at 17. While the second connecting device can certainly "include" other structures and circuits (and must include at least an analog-to-digital converter and a sampling circuit), it would be improper not to give meaning to the term "second connecting device" as well. *See, e.g.*, *Mangosoft*, 525 F.3d at 1330 (Fed. Cir. 2007) (construction that rendered term superfluous was improper).

Papst's construction of "second connecting device" in the '449 patent fares no better. In the '449 patent, Papst offers a proposed construction of the term (rather than attempting to eliminate it), but Papst does so by proposing a vague and overly broad definition inconsistent with the ordinary meaning—it construes the "second connecting device" as a "circuit device" that "couples." Papst Br. at 27-28 This too is incorrect. The connecting device must have a plug or socket, not just a circuit. *See, e.g.*, '399, col. 9:30-34; 9:49-64; Fig. 2 (describing connectors). Papst's attempt to read the connector out of the "*second connecting device*" must be rejected.

G. *"first command interpreter"*

PROPOSED CONSTRUCTION: A software program for interpreting an inquiry from a host device and sending a signal to the host device in response to the inquiry that "lies to the host computer as to the real nature of the data transmit/receive device."

The "*first command interpreter*" is described in the '399 patent, claim 1:

> wherein the interface device is configured by the processor and the memory to include a first command interpreter and a second command interpreter, wherein the first command interpreter is *configured in such a way* that the command interpreter, *when receiving an inquiry* from the host device as to a type of a device attached to the multi-purpose interface of the host device, *sends a signal, regardless of the type of the data transmit/receive device attached* to the second connecting device of the interface device, to the host device which signals to the host device *that it is an input/output device customary in a host device,…* '399: col. 12:61- col. 13:5 (emphasis added).

Because the *first command interpreter* is expressed as being "include[d]" in the interface device

as a result of "configur[ing]" by the processor and memory, and that it is "configured in such a way" to respond to an inquiry, the *first command interpreter must be* a software program or module. Then, when receiving an inquiry from the host device, the *first command interpreter* sends a signal in response, which necessarily requires that it must have interpreted the inquiry to determine what signal to send. This part of the construction appears to be undisputed by Papst. *See* Papst Br. at 18.

But, as properly construed, the *first command interpreter* also "lies to the host computer as to the real nature of the data transmit/receive device." The prosecution history establishes that the "*first command interpreter*" does more than simply indicate that the attached data transmit/receive device is "customary in a host device." After the patent examiner rejected all of the pending claims in view the Applicant's admitted prior art and U.S. Patent No. 5,499,378 to McNeill, Jr. et al. ("McNeill"), the Applicant argued:

> In addition, this reference [i.e., McNeill] does not include a first command interpreter that, when asked by the host device as to the type of device connected to the interface, *lies to the host computer as to the real nature of the data transmit/receive device*. In McNeill, Jr. et al., the initiator asks for a hard disk and the target states that there is a hard disk. *In other words, the target does not lie as to the true type of the data transmit/receive device*. The conversion done in McNeill; Jr. et al. is to transfer SCSI protocol commands into commands understandable for the magnetic disk.

Ex. C, '399 File History, March 18, 2002 Response at 6 (emphasis added). Accordingly, the Applicant unequivocally distinguished McNeill by explaining that the system in McNeill does *not* lie as to the real nature of the device being attached, while the claimed invention does. This argument to distinguish the amended claims from the McNeill reference represents an express, clear, and unambiguous disclaimer applicable to the "*first command interpreter*" limitation. *See, e.g., Solomon Technologies, Inc. v. Int'l Trade Commission*, 524 F.3d 1310, 1313-14 (Fed. Cir. 2008); *Verizon Services Corp. v. Vonage Holdings Corp.*, 503 F.3d 1295, 1306-07 (Fed. Cir.

2007). Papst cannot now seek a construction that tries to recapture the scope of the claims that was expressly disclaimed in order to obtain allowance of the patent. As a result, the "*first command interpreter*" when responding to the host device's inquiry must "lie[] to the host computer as to the real nature of the data transmit/receive device."

In its Brief, Papst suggests that the Applicant's statements attempting to distinguish the McNeill reference are merely in agreement with the claim language that the host computer will not attempt to find or use a specific software driver no matter what data transmit/receive device is attached and that the data transmit/receive device does not factor into the response to the inquiry. Papst Br. at 19. Papst is incorrect on both counts. First, the "*first command interpreter*" lying to the host about the real nature of the data transmit/receive device and indicating that the attached device is an input/output device customary in a host device implies nothing about which device driver the host computer will attempt to find or use.

Second, to "lie" to the host about the real nature of the data transmit/receive device, the "*first command interpreter*" must factor into its response the nature of the data transmit/receive device. It is axiomatic that if the "*first command interpreter*" does not recognize what the data transmit/receive device is, it cannot lie about it.

H. *"second command interpreter"*

PROPOSED CONSTRUCTION: A software program for translating data request commands from the host into data transfer commands understandable by a plurality of dissimilar data transmit/receive devices.

As described in the claims, the interface device is configured to include a first command interpreter and a second command interpreter. '399, col. 12:61-63. Still further, the second command interpreter is "configured to interpret a data request command from the host device … as a data transfer command for initiating a transfer of the digital data to the host device." *Id.* at col. 13:9-13.

The specification explains that the interface device carries out two sets of commands. One set of commands comes from the host. '399, col. 5:64-67 ("Communication between the host system or host device and the interface device is based on known standard access commands as supported by all known operating systems (e.g. DOS, Windows, Unix)."). The first command interpreter interprets those commands. The second command interpreter "carries out the read/write assignment to specific functions" to initiate data transfer from the data transmit/receive device. *Id.* at col. 6:52-67. In the example described by the specification, the host system sends a command to the interface device such as "read file xy." *Id.* The second command interpreter in the interface device interprets this read command as a data transfer command, and begins to transfer data from the data transmit/receive device to the host. *Id.* Thus, the second command interpreter translates data request commands from the host (e.g., "read file xy") into commands understandable by any of the plurality of dissimilar data transmit/receive devices then-attached to the interface device.

I. "*multi-purpose interface*"

PROPOSED CONSTRUCTION: A communication interface designed for use with multiple devices having different functions from each other.

Papst defines "*multi-purpose interface*" as "any interface to which more than one type of device may be attached." Papst Br. at 24. However, to be "multi-purpose," the interface must not only be capable of connecting to and communicating with multiple types of devices, but also devices having more than one purpose, i.e., having different functions from each other. *See, e.g.*, '399, col. 4:40-60; *see also* '399 patent, col. 5:22-32 and 11:9-22 (describing multi-purpose interfaces as being interfaces such as SCSI that are configured to connect to and communicate with multiple types of devices having different functions). Accordingly, Papst's construction of "*multi-purpose interface*" is incorrect.

J. *"interfacing"*

PROPOSED CONSTRUCTION: Physically connecting.

The specification is consistent with the Camera Manufacturers' proposed interpretation of "interfacing" as "physically connecting." Specifically, the first connecting device is identified as a physical connection between the host device and the interface device via connector pins. '399, col. 9:30-48; *see also supra*, "first connective device." Similarly, the specification also identifies the second connecting device as a physical connection between the interface device and the data transmit/receive device via connector pins:

> Preferably, the second connecting device comprises 8 BNC inputs with the calibration relay 1505, a block 1510 with 8 device amplifiers with an overvoltage protection of .±.75 V, this block being connected in turn to 8 sample/hold (S&H) circuits 1515.

'399, col. 9:49-53; *see also supra*, "second connecting device." No other form of interfacing besides physical connection is disclosed or suggested in the specification.

K. *"wherein the interface device is configured by the processor and memory to include a first command interpreter and a second command interpreter"*

PROPOSED CONSTRUCTION: The processor of the interface device (1) runs a program from its memory to (2) determine the data transfer parameters of the interface device for the first and second command interpreters.

For the processor and memory together to configure the interface device, the processor must necessarily execute software located in its memory to perform the configuration of the interface device. The specification fully supports this part of the construction. As Fig. 1 of the '399 patent depicts, the interface device 10 includes a digital signal processor ("DSP") 13 and a memory means 14. The specification states that the operating system for the DSP 13 is stored in the memory means 14. '399, col. 6:26-27; 10:14-17. It also discloses that an on-chip memory of the DSP, such as RAM 1440, can store certain instruction sets, functions and smaller application software units for the DSP. '399, col. 10:23-27. Thus, the specification requires that the

memory stores a program executed by the processor to perform configuring functions for the interface device.

The second part of the proposed construction is based on the inherent feature of the claimed interface device that it is configured to be attachable to different types of data transmit/receive devices for communication with a host device. '399, col. 6:19-22, 8:23-31. As discussed herein, the host device does not know and need not recognize what type of data transmit/receive device is attached to the interface device. *Id.* So that the host device and the data transmit/receive device can communicate, the interface device handles the task of recognizing different data transmit/receive devices, and then is configured to facilitate communication between the host and the various data transmit/receive devices.

To facilitate that communication, the interface device includes first and second command interpreters, which are present regardless of the type of data transmit/receive device that is attached to the interface device. '399, col. 6:52-67. What does vary depending on the type of data transmit/receive device that is attached are the settings in these command interpreters that enable communication between the host and a particular data transmit/receive device. *Id.* These data transfer settings "configure" the interface device according to the data transmit/receive device that is attached.

L. ***"sends a signal, regardless of the type of the data transmit/receive device attached to the second connecting device of the interface device, to the host device which signals to the host device that it is an input/output device customary in a host device"***

PROPOSED CONSTRUCTION: The first command interpreter sends a signal to the host device that the data transmit/receive device is an input/output device customary in a host device, and in so doing "lies to the host computer as to the real nature of the data transmit/receive device.

Papst's proposed construction starts by limiting the term "signal" to that which is "[a]t least with respect to the portion of the signal relevant to software driver selection." Papst Br. at

18. However, there is no support in the specification to support such a limitation. Furthermore, Papst's suggestion that the construction of this phrase includes the language "signal is consistent with a signal that an input/output device customary in a host device would provide in response to the inquiry, and that such as [sic] response is not based on what data transmit/receives devices may be associated with the interface device" (Papst Br. at 18-19) focuses as much on the "response" to the signal as it does the signal. Any construction for "sends a signal" should concern the signal itself, not some vague reference to an unclaimed response.

As explained in connection with the "*first command interpreter*," the specification and prosecution history require the interface device to "lie[]" to the host device as to the real nature of the data transmit/receive device. The Camera Manufacturers' construction accounts for the meaning of each term, is consistent with the claim language, specification, and prosecution history, and avoids unnecessary and unhelpful use of the claim terminology in the construction. Because Papst's construction ignores the intrinsic evidence and is a largely non-informative regurgitation of the language to be construed, the Camera Manufacturers' construction of this term should be adopted.

M. "*the driver*"

PROPOSED CONSTRUCTION: The set of software routines that control an input/output device.

The term "*driver*" is recited in claims 1, 2, 11, 14, and 15 of the '399 patent, and in claims 1, 17, and 18 of the '449 patent. The Camera Manufacturers' construction comports with the well understood meaning of the term "*driver*" to one of ordinary skill in the art at the time of the alleged invention. IEEE defines the term "*driver*" as a computer program, i.e., a set of software routines, that controls a peripheral device and reformats data for transfer to and from the device. *See* Ex. G, THE NEW IEEE STANDARD DICTIONARY OF ELECTRICAL & ELECTRONICS

TERMS (5th ed.) (1993). Similarly, IEEE defines the term "device driver," a term that as Papst admits is equivalent to "*driver*" (*see* Papst Br. at 1), as the software that manages I/O operations for a particular device, contains all the device-specific code necessary to communicate with the device and provides a standard interface to the rest of the system. *See* Ex. H, IEEE 100 THE AUTHORITATIVE DICTIONARY OF IEEE STANDARDS TERMS (7th ed.) (2000). According to these definitions, to be a driver, software must perform the function of controlling a device. Accordingly, these definitions clearly support the Camera Manufacturers' construction.

The specification also fully supports this meaning of the term "*driver*." For example, the specification repeatedly discusses drivers as the software mechanism for allowing a host device to communicate with and control an interface device. '399, col. 2:23-63. The specification also similarly describes the functions of drivers:

> As described above, *communication between the host device and the multi-purpose interface can take place* not only via drivers for input/output device customary in a host device which reside in the BIOS system of the host device but also via specific interface drivers which, in the case of SCSI interfaces, are known as multi-purpose interface ASPI (advanced SCSI programming interface) drivers. This ASPI driver, which can also be referred to as an ASPI manager, is specific to a special SCSI host adapter, i.e. to a special multi-purpose interface, and is normally included by the manufacturer of the multi-purpose interface. Generally speaking, *this multi-purpose interface driver has the task of moving precisely specified SCSI commands from the host system program to the host system SCSI adapter*. For this reason, the command set is almost identical to that of the SCSI interface itself.

'399, col. 11:9-24. The above recited description confirms that "*the driver*" cannot merely be a software element of a driver which by itself is useless, but must be *the set* of software routines that have overall functionality of controlling a device. In the context of the patents-in-suit, the driver must be what actually enables the host device to communicate with and control an input/output device.

N. *"the driver for the input/output [storage] device customary in a host device"*

PROPOSED CONSTRUCTION: The driver normally present in most commercially available computers at the time of the invention.

Independent claim 1 of the '399 patent recites "the driver for the input/output device customary in a host device" and independent claim 1 of the '449 patent recites "the driver for the storage device customary in a host device." The patentee explicitly defined these phrases in the specification to mean "the driver normally present in most commercially available host devices":

> The present invention is based on the finding that both a high data transfer rate and host device-independent use can be achieved if *a driver for an input/output device customary in a host device, normally present in most commercially available host devices*, is utilized. Drivers for input/output devices customary in a host device which are found in practically all host devices are, for example, drivers for hard disks, for graphics devices or for printer devices.

'399, col. 4:23-30 (emphasis added). Because the patentee acted as his own lexicographer, "the inventor has dictated the correct claim scope, and the inventor's intention, as expressed in the specification, is regarded as dispositive." *Phillips*, 415 F.3d at 1316. Further, because the patentee has acted as his own lexicographer, Papst's reliance on extrinsic evidence is wholly improper. *Phillips*, 415 F.3d at 1316, 1318; *Bell & Howell Doc. Mgmt. v. Altek Sys.*, 132 F.3d 701, 706 (Fed. Cir. 1997) ("When the intrinsic evidence is unambiguous, it is improper for the court to rely on extrinsic evidence such as expert testimony for purposes of claim construction.").

Moreover, claim terms are to be interpreted to have the meaning they had at the time of the invention, i.e., as of the effective filing date of the patent application. *Phillips*, 415 F.3d at 1313; *see also PC Connector Solutions LLC v. SmartDisk Corp.*, 406 F.3d 1359, 1363 (Fed. Cir. 2005) ("A claim cannot have different meanings at different times; its meaning must be interpreted as of its effective filing date."). The Federal Circuit reaffirmed this precedent last week holding that a "conventional" Internet browser is limited to web browsers in existence at

the time of the alleged invention. *See Muniauction, Inc. v. Thomson Corp.*, __ F.3d __, 2008 WL 2717689 at *5 (Fed. Cir. July 14, 2008). Thus, certain terms, such as "normally," "conventional," "traditionally," and "standard," which are implicitly time-dependent, necessarily have a meaning specific to the time of filing. *PC Connector*, 406 F.3d at 1363-64 (limiting "input/output port normally connectible to a conventional computer input/output port" and "standard input/output port" that is "traditionally connectable to a computer" to input/output technologies existing at the time of filing). *See also Kopykake Enterprises, Inc. v. Lucks Co.*,, 264 F.3d 1377, 1384 (Fed. Cir. 2001) (holding that "conventional screen printing" did not cover ink jet printing because that was not a conventional method of printing at the time of filing). Thus, the claim term "customary" (and the word "normally" as used by the patentee to define "customary") in the '399 and '449 patents is a time-dependent term that must be construed to refer to only what was "customary" at the time of the alleged invention. Consequently, Papst's proposed construction, which seeks to construe the term "customary" to encompass technology that would be "customary" or "normally used" today, is improper.

Finally, without any support, Papst argues that the term "customary" in this limitation modifies the term "input/output device," not the term "drivers." Papst Br. at 20, 29. But the patent makes clear that the phrase "customary in a host device" modifies "drivers," not the prepositional phrase "for input/output devices:"

> Drivers for input/output devices customary in a host device which are found in practically all host devices are, for example, drivers for hard disks, for graphics devices or for printer devices.

'399, col. 4:27-30. While the recited drivers for hard disks, for graphics devices and for printers are customary in a host device, not all of the recited devices, such as printers, which are external to a computer, are customary "in" a computer. Accordingly, the phrase "customary in a host

device" cannot modify the phrase "input/output devices," which include printers, as argued by Papst.

O. *"an input/output [storage] device customary in a host device"*

PROPOSED CONSTRUCTION: A data input/output [storage] device that was normally present within the chassis of most commercially available computers at the time of the invention. The I/O devices that were customary in personal computers were hard disk drives, floppy disk drives, CD-ROM drives and tape drives.

Independent claims 1, 11 and 14 of the '399 patent each recite "an input/output device customary in a host device" and independent claims 1, 17 and 18 of the '449 patent each recite a "storage device customary in a host device." The patentee explicitly defined the phrase "customary in a host device" to mean "normally present in most commercially available host devices": "The present invention is based on the finding that both a high data transfer rate and host device-independent use can be achieved if a driver for an input/output device *customary in a host device, normally present in most commercially available host devices*, is utilized." '399, col. 4:23-27 (emphasis added). Because the patentee acted as his own lexicographer and because, as explained above, the claim term "customary" is to be construed as of the date of the alleged invention, "customary in a host device" means "normally present in most commercially available computers at the time of the invention."

Papst's proposed construction is wrong for many of the same reasons that its proposed construction for "the driver for the input/output device customary in a host device" is wrong. In addition, by replacing the claim term "in" with the phrase "with respect to," Papst's proposed construction attempts to read out the term "in" thereby making it superfluous. The Federal Circuit repeatedly has denounced construing claims in a way that would render a claim term superfluous. *See, e.g.*, *Stumbo v. Eastman Outdoors, Inc.*, 508 F.3d 1358, 1362 (Fed. Cir. 2007) (stating that construing a claim in a way that would render a claim term superfluous is a

"methodology of claim construction that this court has denounced"). Accordingly, given its full effect, the term "in" is properly construed to mean that the claimed input/output (and storage) device is located within the chassis of a computer. The patentee identified such input/output (and storage) devices as hard disk drives, floppy drives, CD-ROM drives and tape drives:

> As however the hard disk interfaces in common host devices which can be, for example, IBM PCs, IBM-compatible PCs, Commodore PCs, Apple computers or even workstations, are the interfaces with the highest data transfer rate, the *hard disk* driver is utilized in the preferred embodiment of the interface device of the present invention. Drivers for *other storage devices such as floppy disk drives, CD-ROM drives or tape drives* could also be utilized in order to implement the interface device according to the present invention.

'399, col. 4:30-39 (emphasis added). Contrary to Papst, printers are not, and the patent does not disclose that they are, input/output devices "in" a computer. Papst's proposed construction is also improper because it adds the extraneous limitation of software drivers to the claim term "input/output device."

P. *"specific driver for the multi-purpose interface"*

PROPOSED CONSTRUCTION: The set of software routines that control the multi-purpose interface that are developed for the particular multi-purpose interface.

The multi-purpose interface referenced in claim 11 of the '399 patent and claim 17 of the '449 patent is a part of the host device. *E.g.,* '399, col. 14:7-8. The specific driver for that multi-purpose interface enables the host device to communicate with and control the multi-purpose interface. The modifier "specific" is used in the patents-in-suit to refer to the fact that driver is not a general driver or one which is customary in the host device, but rather one that is developed for the particular multi-purpose interface employed in the host device:

> As described above, communication between the host device and the multi-purpose interface can take place not only via drivers for input/output device customary in a host device which reside in the BIOS system of the host device but also via specific interface drivers which, in the case of SCSI interfaces, are known

> as multi-purpose interface ASPI (advanced SCSI programming interface) drivers. This ASPI driver, which can also be referred to as an ASPI manager, is specific to a special SCSI host adapter, i.e. to a special multi-purpose interface, and is normally included by the manufacturer of the multi-purpose interface.

'399, col. 11:9-19.

Papst's proffered construction—"a software driver that enables a host system to communicate via a multipurpose interface" (Papst Appendix at 5)—is wrong because it ignores the explicit requirement that the driver must be *specific* to the particular multi-purpose interface employed in the host device.

Papst's construction is also wrong because "*the specific driver*" cannot merely be a software element of a driver that enables communications:

> In general terms, the ASPI manager comprises two sides. One side is the proprietary, hardware-oriented side. It is responsible for converting all commands into a form required by the corresponding multi-purpose interface. The hardware-oriented side of the ASPI driver is therefore matched to a very specific type of multi-purpose interface or SCSI interface. The other side is known as the user software side. This side is totally independent of the proprietary operating characteristics of the SCSI adapter and is therefore identical for all SCSI interfaces. This permits SCSI programming which is however independent of the individual SCSI adapter types.

'399, col. 11: 31-42. Thus, contrary to Papst's construction, the *specific driver* must be a complete *set* of software routines that have overall functionality of controlling a device.

Q. ***"whereupon the host device communicates with the interface device by means of the driver for the input/output [storage] device customary in a host device"***

PROPOSED CONSTRUCTION: No construction is necessary for the entire limitation; the parties have separately construed "host device," "interface device" and "driver for the input/output device customary in a host device."

Independent claim 1 of the '399 patent recites the limitation "whereupon the host device communicates with the interface device by means of the driver for the input/output device customary in a host device." Independent claim 1 of the '449 patent recites a similar limitation, replacing the term "input/output device" with "storage device."

Papst seeks to construe the entire limitation even though the parties have proposed constructions for its constituent claim terms (e.g., "host device," "interface device" and "driver for the input/output [storage] device customary in a host device."). Consequently, it is unnecessary and improper to construe the entire limitation.

Moreover, Papst's proposed construction improperly seeks to read in a limitation that the host device "automatically" uses a driver. Papst fails to provide any intrinsic evidence to support its proposed construction or for the addition of the term "automatically." Not only is Papst's construction unsupported, it also creates unnecessary ambiguity (i.e., what does it mean for a host device to "automatically" use a driver?).

Papst's construction is also inconsistent with its on-going prosecution of related patent applications that explicitly recite an "automatically" limitation—in contrast to the claims here. *See, e.g.*, Ex. I, Appl'n No. 11/078,778, Preliminary Amendment, Mar. 28, 2006, claim 17 at 4. Although Papst is drafting the claims of its pending applications to explicitly recite the "automatically" limitation, it chose not to do so for the claims of the '399 and '449 patents. Papst's after-the-fact attempt to read in an "automatically" limitation to these patents is improper. *See Mycogen Plant Science v. Monsanto Co.*, 243 F.3d 1316, 1328 (Fed. Cir. 2001) (rejecting proposed construction that read an additional limitation into the claims stating that had the patentee meant the claims to require such a limitation, the patentee could have and should have included the limitation in the claims themselves).

R. *"the digital data"*

PROPOSED CONSTRUCTION: The same digital data output from the analog-to-digital ("A/D") converter, unmodified by additional processing.

It is a straightforward matter of claim construction that definite articles such as "the" and "said" are used to refer back to elements previously identified in a claim. *See, e.g., Process*

Control Corp. v. HydReclaim Corp., 190 F.3d 1350, 1356-57(Fed. Cir. 1999) (holding "the discharge rate" means the "same" rate previously mentioned in the claim); Ex. J, MANUAL OF PATENT EXAMINING PROCEDURE § 2173.05(e) (indicating that either the term "said" or "the" should precede a claim limitation that has an antecedent basis). In the context of the '399 patent claims, "the digital data" refers back to the *same* digital data first referenced in each claim: the digital data output by the A/D converter, unmodified by additional processing. For example, claim 1 of the '399 patent recites first an "analog-to-digital converter for converting data sampled by the sampling circuit into *digital data*" and then a "second command interpreter for initiating a transfer of *the digital data* to the host device." '399 patent, cols. 12:42-13:12 (emphasis added).

Moreover, the Federal Circuit recently affirmed a construction of remarkably similar claim language. In *Ampex Corp. v. Eastman Kodak Co.*, the court granted summary judgment that Kodak's digital cameras did not infringe the asserted patents based on the relationship between the first reference to "data" and subsequent references to "said [ ] data" and "the data" in each of the claims asserted in that case. Ex. K, *Ampex Corp. v. Eastman Kodak Co.*, 460 F. Supp. 2d 563 (D. Del. 2006), *aff'd*, 2008 WL 343253 (Fed. Cir. 2008). In *Ampex*, the District Court agreed with defendants that "the data" and "said data" had to be the same numerical information as the earlier "data" in the claims. 460 F. Supp. 2d at 565-66. As a result, the defendants' cameras did not literally infringe the asserted patents because after receiving the earlier "data," the cameras processed that data (*e.g.*, by performing compression, color correction, etc.) such that the processed numerical information—which the plaintiff alleged corresponded to "the/said data"—was different from the original numerical information. 460 F. Supp. 2d at 565-66. The Federal Circuit affirmed this decision. *See* 2008 WL 343253 (Fed. Cir. 2008).

During prosecution of the '399 patent, Tasler confirmed that "the digital data" is the same digital data output from the A/D converter:

> [T]he second command interpreter interprets a data request command from the host device . . . as a data transfer command for *initiating a transfer of the digital data* to the first device, i.e., of *the digital data that have been derived from the analog data of the data transmit/receive device*.

See Ex. C, '399 File History, March 18, 2002 Response at 5. Moreover, Tasler was forced to add the "digital" and "the digital data" language to the claims to overcome an obviousness rejection over prior art. *See* Ex. C, '399 File History, March 18, 2002 Response at 2-6; Ex. L Notice of Allowability. To allow Papst essentially to ignore the "*the* digital data" limitation by construing it to mean data other than the same digitized analog data output from the A/D converter would allow Papst to maneuver around exactly what the PTO required to be added to the claims to make them, in the PTO's view, non-obvious and therefore patentable over the cited prior art. *See Lemelson v. General Mills Inc*., 968 F.2d 1202, 1207-08 (Fed. Cir. 1992) ("Lemelson cannot acquiesce to a rejection and to an agreed alternative [amending claims in light of an obviousness rejection], and now years later shift his stance 180° to argue for a second bite at the abandoned apple").

S. *"inquiry"* [or *"inquiring"*]

PROPOSED CONSTRUCTION: The SCSI inquiry command [or sending the SCSI inquiry command]

Independent claims 1, 11 and 14 of the '399 patent and independent claims 1, 17 and 18 of the '449 patent each recite that the host device sends an inquiry to the interface device to determine the type of device attached to the host device. The patentee acted as his own lexicographer stating that the claimed inquiry is the specific inquiry instruction known by those skilled in the art:

When the host device system with which the interface device according to the

> present invention is connected is booted and a data transmit/receive device is also attached to the interface device 10, usual BIOS routines or multi-purpose interface programs issue *an instruction, known by those skilled in the art as the INQUIRY instruction*, to the input/output interfaces in the host device. The digital signal processor 13 receives *this inquiry instruction* via the first connecting device and generates a signal which is sent to the host device (not shown) again via the first connecting device 12 and the host line 11. This signal indicates to the host device that, for example, a hard disk drive is attached at the interface to which *the INQUIRY instruction* was sent.

'399, col. 6:3-16 (emphasis added, capitalization in original). This known inquiry instruction is the SCSI inquiry command, which is one of the commands for the SCSI ("small computer system interface") standard described in the patent for connecting and transferring data between a computer and a peripheral. *See, e.g.*, '399, col. 9:29-37; 11:19-25 (referencing SCSI commands); *see also* Ex. M, *American National Standard for Information Systems – Small Computer System Interface – 2* at 96-100 (1994) (describing the SCSI inquiry command). Papst's proposed construction ignores the explicit definition of the claim term "inquiry." By acting as his own lexicographer, the patentee's definition is dispositive and Papst's reliance on extrinsic evidence is wholly improper. *Phillips*, 415 F.3d at 1316, 1318; *Bell & Howell*, 132 F.3d at 706.

Independent method claim 14 of the '399 patent and independent method claim 18 of the '449 patent each recite the step of "*inquiring* by the host device at the interface device as to the type of device to which the multi-purpose interface of the host device is attached." The subsequent limitations of these claims recite that the interface device responds to "the inquiry" from the host device. Accordingly, the claimed "inquiring" step requires the host device to send the inquiry command discussed above to the interface device. Thus, the term "inquiring" should be construed to mean "sending the SCSI inquiry command."

T. '399 Patent, Claim 2

PROPOSED CONSTRUCTION: No construction needed

Claim 2 of the '399 patent recites "An interface device according to claim 1, wherein the drivers for input/output drivers customary in a host device comprise a hard disk driver, and the signal indicates to the host device that the host device is communicating with a hard disk." Each of the disputed terms in this claim are already being construed in the context of independent claim 1 (*e.g.*, interface device, the driver, host device, and customary), and there is no need to construe any of the remaining claim language.

Papst's "construction" does nothing to clarify the claim language, and is simply a rewrite of the entire claim:

> Claim 2 should be construed to require the interface device to send a signal in response to the inquiry from the computer that indicates to the computer that the interface device is a particular type of input/output device, i.e., a hard disk drive, and that the computer communicate with the interface device using a driver for a particular type of input/output device, i.e., a hard disk drive.

Papst Br., Appendix at 3. It is improper to rewrite a claim under the guise of claim construction. *Chef America, Inc. v. Lamb-Weston, Inc.*, 358 F.3d 1371, 1374 (Fed. Cir. 2004) (Courts "construe the claim as written, not as the patentees wish they had written it.").

Furthermore, Papst's construction improperly deletes an important claim limitation: "customary." *Merck & Co. v. Teva Pharm. USA, Inc.*, 395 F.3d 1364, 1372 (Fed. Cir. 2005) ("A claim construction that gives meaning to all the terms of the claim is preferred over one that does not do so."); *Elekta Instrument S.A. v. O.U.R. Scientific Int'l, Inc.*, 214 F.3d 1302, 1307 (Fed. Cir. 2000) (construing a claim to avoid rendering the 30 degree claim limitation superfluous).

U. "a buffer to buffer data to be transferred between the data transmit/receive device and the host device" or "a data buffer for permitting independence in terms of time of the data transmit/receive device attachable to the second connecting device from the host"

PROPOSED CONSTRUCTION: Volatile memory used to temporarily store data to compensate for differences between the rate in the flow of data between the data transmit/receive device and the host device.

Papst treats these terms indistinguishably. The first appears in claim 3 of the '399 patent (which depends from claim 1), and the second appears in claim 16 of the '449 patent (which depends from claim 1). Papst proffers the same construction for each term—the "memory [of claim 1] is adapted to store the data gathered by the transmit/receive device until it is transferred . . . ." Papst Br. at 22. Under Papst's proposed construction, the data could be stored indefinitely. That would make "buffer" in '399, claim 3 and `449, claim 16 indistinguishable from "memory" in claim 1 of the respective patents, giving '399, claim 3 the same scope as '399, claim 1, and likewise for '449, claim 16.

Papst's proposal violates the doctrine of claim differentiation, which creates a presumption that different claims have different scopes. *See Phillips* , 415 F.3d at 1314-15 ("The presence of a dependent claim that adds a particular limitation gives rise to the presumption that the limitation in question is not present in the independent claim."). Papst's proposed construction also violates 35 U.S.C. § 112 ¶ 4, which requires that dependent claims include a "further limitation" — meaning a dependent claim must have narrower scope than the independent claim from which it depends. 35 U.S.C. § 112 ¶ 4 ("[A] claim in dependent form shall contain a reference to a claim previously set forth and then specify a <u>further limitation</u> of the subject matter claimed" (emphasis added)).

In the context of Papst's '399 and '449 patents, a buffer is used in its plain and ordinary sense, which is a particular kind of memory, namely a temporary one. As Mr. Tasler has

explained, "buffers have to be provided [] for *temporary* storage of [] data." *See* Ex. N, Michael Tasler, *Design and Construction of a Universal Data Acquisition and Control System for Scanning Probe Microscopy* at 15 (1996) (emphasis added). More specifically, a "buffer" in its plain and ordinary sense is "[a] *temporary* memory for data, normally used *to accommodate the difference in the rate* at which two devices can handle data *during a transfer*." Ex. O, Definition of "Buffer" in the OXFORD DICTIONARY OF COMPUTING (4th ed. 1996) (emphasis added).

In its plain and ordinary sense, a buffer is not only temporary, but it is also "volatile," meaning it stores information only when it is powered. *Id*. (stating that a buffer is used "*during* a transfer"). Every single buffer described in Papst's '399 patent is described as random access memory ("RAM"), which is a volatile memory. *See, e.g.*, the buffer 1420 shown in Figure 2; col. 10:18-23 ("A random access memory with an access time of 15 ns and a size of 512 KB or optionally 1024 KB 1420 serves as a data buffer to achieve independence in terms of time of the output line 16 [to the data transmit/receive device] from the output lines 11a, 11b and 11c [] to the host device."); Col. 7:30-31 ("Preferably, the buffer is implemented as a fast random access memory or RAM buffer.").

Moreover, a buffer is normally used between two devices to synchronize the transfer of data between them, or in other words, to account for the difference in the rate at which one device can send data and another can receive data. Ex. O, Definition of "Buffering" in the OXFORD DICTIONARY OF COMPUTING. The OXFORD DICTIONARY OF COMPUTING explains how a buffer is normally used:

> [B]uffering [is] used to compensate for the slow and possibly erratic rate at which a peripheral device produces or consumes data. Consider a program [*e.g.*, on a data transmit/receive device] sending output to a slow device [*e.g.*, a host device]. A memory area (the buffer) is set aside for communication: the program places data in the buffer at its own rate, while the device takes data from the buffer at its own rate. Although the device may be slow, the program does not have to stop

> unless the buffer fills up; at the same time the device runs at full speed unless the buffer empties.

Id. The buffer as used in Papst's '399 patent is used no differently than an ordinary buffer as described. Similarly, the specification explains that a buffer is used "for purposes of synchronizing data transfer from the data transmit/receive device to the interface device 10 and data transfer from the interface device 10 to the host device." '399, col. 7:26-29.

V. *"virtual files"*

PROPOSED CONSTRUCTION: A file that does not physically exist as a file in the interface device but appears to the host device to be an actual file, and references data to be transmitted between the data transmit/receive device and the host device.

Claim 7 of the '399 patent recites an interface device "which further comprises a root directory and *virtual* files which are present on the signaled hard disk drive and which can be accessed from the host device." Papst bases its constructions of "virtual file" on the incorrect, and wholly unsupported, premise that *actual* files and file systems can only exist on rotating media. Papst Br. at 23 ("virtual files should be construed to mean files that appear to be stored on an emulated disk drive, but are not actually on a rotating disc."). On the contrary, actual files may exist on any type of media, including hard drives, floppy drives, flash media, CD-ROM drives, etc.

The crux of the purported invention is to take an analog stream of data from any number of different data transmit/receive devices and make that analog data stream appear to a host device as just another file stored somewhere on a storage device, which it can access as it would any other file. *See* Ex. C, '399 File History, March 18, 2002 Response at 5; '399 Patent, 5:6-20. It is irrelevant what hardware the interface device uses to support the ruse; the point is that it appears to the host as a file, not that it looks like it is on rotating media.

As has been discussed thoroughly above, the interface device presents itself as a hard

disk to the host device. '399, col. 6:58-59. When the host device sends a "read" command to the interface device, the host thinks it is reading a file, and the interface device maintains this ruse by transferring data from the data transmit/receive device to the host device in the form of a "real-time input" file. *Id.*, 6:59-67. "[T]he 'real-time input' file then appears [to the host device] as a file whose length corresponds to the anticipated volume of data" specified in a configuration file. *Id.*, 7:1-7. Thus, the "real-time input" file is real-time streaming data from the data transmit/receive device, reformulated by the interface device to appear to be a file. These are the virtual files referenced in the specification.

The terms "virtual" and "virtual file" are well understood by those of ordinary skill in the art, and there is no evidence in the intrinsic record that Mr. Tasler meant to use these terms in any unique way. The specification's description of virtual files is consistent with the understanding of people of ordinary skill in the art in the relevant time period, as evidenced by technical dictionaries from that period. The New IEEE Dictionary of Electrical & Electronics Terms (5th ed.) 1993 defines the term "virtual record" as "[a] record that appears to be but is not physically stored; rather, it is constructed or derived from existing data when its contents are requested by an application program." *See* Ex. G. This definition is practically identical to the way the term "virtual real-time input file" is used in the specification.[3]

Moreover, Papst's assertion that this construction somehow "exclude[s] files that are saved in memory on the interface device," (Papst Br. at 23, citing claim 9 of the '399 patent)

---

[3] Other contemporary dictionaries are also consistent. *See, e.g.*, Ex. P, OXFORD ENGLISH DICTIONARY (2d Ed. 1989) (defines "virtual" in the context of computers as "Not physically existing as such but made by software to appear to do so from the point of view of the program or the user"); Ex. Q, S.M.H. Collin, DICTIONARY OF PERSONAL COMPUTING AND THE INTERNET (1997) (defines "virtual" as "something that does not actually exist, except in an imaginary form in a computer").

ignores the context of the claim language. Claim 7, from which claim 9 depends, says that "virtual files … are present *on the signaled hard disk drive*." Papst admits that the signaled hard disk drive is not real, but rather refers to a signal "that indicates that a hard disk drive has been attached to the computer (when that is not actually the case)." Papst Br. at 24. In other words, the virtual files in claim 7 (and claim 9) "are present on the signaled hard disk drive," which is meant to appear to the host device to be on the data transmit/receive device but does not actually exist. Thus, "the batch files or executable files… which are stored in the interface device" recited in claim 9 are "virtual" in the sense that they are made to appear on the data transmit/receive device, but do not.

Moreover, the specification does not explain how the batch or executable files from claim 9 are stored and implemented on the interface device, nor does it refer to them to "virtual" files. If claim 9 were asserted here—which it is not—the Camera Manufacturers would argue that it is invalid for lack of sufficient written description. 35 U.S.C. § 112. Nonetheless, it does appear from the specification that the batch or executable files are meant to appear to the host as though they are on the data transmit/receive device. The hard drive that Papst admits doesn't exist in the first place, doesn't become more real just because Papst claims to have stored batch and executable files there. The data transmit/receive device's virtual files and signaled hard disk are emulated, and it is irrelevant what hardware on the interface device is used to implement them.

W. *"simulating a virtual file system to the host device"*

PROPOSED CONSTRUCTION: Appearing as a file system to the host, but actually maintaining virtual files, where virtual files do not physically exist as files in the interface device but appear to the host device to be actual files and reference data to be transmitted between the host device and the data transmit/receive device.

The Camera Manufacturers maintain that "simulating a virtual file system to the host," which is found in every independent claim of the '449 patent, is indefinite and cannot be

construed. This phrase is ambiguous, vague, nonsensical and not supported by the specification or the prosecution history. To the extent this phrase has any meaning to one skilled in the art, "simulating" is redundant with "virtual" in that "virtual" itself refers to simulation of the structure of an actual file system stored on a hard disk drive or the like. *See* Section V regarding "virtual files" above. There is no support whatsoever in the intrinsic evidence for "simulating" a "virtual file system," which is itself a simulation of an actual file system. Thus, because one skilled in the art would have no way of interpreting this nonsensical claim language, the independent claims of the '449 patent are invalid for indefiniteness. "Where … claims are susceptible to only one reasonable interpretation and that interpretation results in a nonsensical construction of the claim as a whole, the claim must be invalidated." *Chef America, Inc. v. Lamb-Weston, Inc.*, 358 F.3d 1371, 1374 (Fed. Cir. 2004) (internal citations omitted). In addition, the Federal Circuit "repeatedly and consistently has recognized that courts may not redraft claims, whether to make them operable or to sustain their validity." *Id.* If, however, the Court deems this term capable of construction, the Camera Manufacturers submit that it should be construed consistently with the proposed construction of "virtual file" above.

Papst construes a virtual file system as a response "to a host computer with information that is consistent with what would be provided by a disk drive with rotating magnetic media," as if a file system is defined by the underlying physical structure of a storage device. Papst Br. at 30. Papst implies that real file systems can only exist on rotating media, and virtual file systems can only exist on other media. Both conclusions are wrong and unsupported. Indeed, the portion of the specification Papst relies upon for its construction, col. 5:35-44, describes the representation of a virtual hard disk to a host device. It describes how a conventional disk handles a request to display the disk's directory by looking to the file allocation table, or FAT,

"on a sector specified in the boot sequence, normally the first writable sector, and transferring it to the host device." '449, col. 5:35-44. The specification merely notes that rotating media hard disks have sectors; it never suggests that *only* rotating disks have sectors, or that only rotating disks support FAT file systems. The specification does not support Papst's rhetorical leap that somehow a "virtual file system" is one that is "not found on a rotating medium." In contrast, the Camera Manufacturers' proposed construction is consistent with the well-supported construction of "virtual file" discussed in the previous section.

The Camera Manufacturers contend that the term "simulating a virtual file system to the host" is indefinite for the reasons state above. In the event that the Court finds that the term can be construed without rewriting or "fixing" the erroneously recited claim language, then the Court should adopt the Camera Manufacturers' proposed construction.

X. *"the usual driver for the input/output [storage] device"*

PROPOSED CONSTRUCTION: Indefinite; to the extent these claim terms can be construed they should be construed to mean "the driver normally present in most commercially available computers at the time of the invention."

Independent claim 14 of the '399 patent recites "the usual driver for the input/output device" and independent claim 18 of the '449 patent recites "the usual driver for the storage device." These claims are indefinite because there is no antecedent basis for the phrase "the usual driver;" they do not recite any antecedent recitation of "a usual driver." Consequently, these claims are invalid for indefiniteness under 35 U.S.C. § 112, ¶ 2. *See Halliburton Energy Serv., Inc. v. M-I LLC*, 514 F.3d 1244, 1249 (Fed. Cir. 2008) (affirming invalidity for indefiniteness and stating that a claim could be indefinite if no proper antecedent basis).

To the extent these claim terms can be construed, "the usual driver" would refer to the "drivers for input/output devices customary in a host device" recited in the preamble of the claims. Accordingly, "the usual driver for the input/output device" and "the usual driver for the

storage device" should be construed the same as the phrase "drivers for input/output devices customary in a host device" to mean "the driver normally present in most commercially available computers at the time of the invention."

Y. Miscellaneous Terms that Require No Construction

A party's mere assertion that a claim term is "disputed" is insufficient to justify a request that the court construe the term. *See*, *Renishaw PLC v. Marposs Societa' Per Azioni*, 158 F.3d 1243, 1248 (Fed. Cir. 1998). Here, Papst seeks constructions of four terms that the Camera Manufacturers do not believe need to be construed by the Court: "memory," "processor," "root directory," and "digital signal processor." Papst Br. at 16, 22, 23. Papst has never provided any reasons why these terms matter in the litigation, and so the Court should instead focus its energies on terms that currently appear to make a substantive difference in the case. There is no need for the Court to expend its resources defining these terms.

IV. CONCLUSION

For the foregoing reasons, the Court should adopt the Camera Manufacturers' proposed constructions.

DATED: July 23, 2008

By: /s/ Rachel M. Capoccia
Rachel M. Capoccia, Esq.

For The Victor Company of Japan Parties
Rachel M. Capoccia, Esq.
HOGAN & HARTSON LLP
1900 Avenue of the Stars
Suite 1400
Los Angeles, California 90067
Email: rmcapoccia@hhlaw.com
Tel: (310) 785-4744
Fax: (310) 785-4601

For The Matsushita Electric Industrial Co., Ltd. Parties
Richard de Bodo
HOGAN & HARTSON LLP
1900 Avenue of the Stars
Suite 1400
Los Angeles, California 90067
Email: rdebodo@hhlaw.com
Tel: (310) 785-4694
Fax: (310) 785-4601

For The Casio Parties
J. Kevin Fee
MORGAN, LEWIS & BOCKIUS LLP
1111 Pennsylvania Avenue, NW
Washington, D.C. 20004
jkfee@morganlewis.com

For The Samsung Parties
Patrick J. Kelleher
Drinker Biddle & Reath LLP
191 North Wacker Drive
Suite 3700
Chicago, IL 60606-1698
Phone: (312) 569-1375
Fax: (312) 569-3375
patrick.kelleher@dbr.com

For the Fujifilm parties
Steven J. Routh
Sten A. Jensen
Orrick, Herrington & Sutcliffe LLP
Columbia Center
1152 15th St, NW
Washington DC 20005
Phone: (202) 339-8400
Fax: (202) 339-8500
srouth@orrick.com
sjensen@orrick.com

For the Olympus Parties
Richard de Bodo
David H. Ben-Meir
Hogan & Hartson, LLP
1999 Avenue of the Stars
Suite 1400
Los Angeles, CA 90067
Phone: (310) 785-4694

Fax: (310) 785-4601
rdebodo@hhlaw.com

For Hewlett-Packard Company:
Heather N. Mewes
Fenwick & West LLP
555 California Street, Suite 1200
San Francisco, CA 94104
Phone: (415) 875-2300
Fax: (415) 281-1350
hmewes@fenwick.com

For Ricoh Americas Corp.
Paul Devinsky
McDermott Will & Emery LLP
600 13th Street N.W.
Washington D.C. 20005-3096
Tel- 202.756.8000 (Main)
Fax -202.756.8087
Mobile-301.512.0060
pdevinsky@mwe.com

For the Nikon Parties
David L. Witcoff
Jones Day
77 W. Wacker Drive
Chicago, Illinois 60601
Direct: 312-269-4259
Fax: 312-782-8585
E-mail: dlwitcoff@jonesday.com

CERTIFICATE OF SERVICE

I, Rachel M. Capoccia, hereby certify that on this 23rd day of July 2008, a true and correct copy of the foregoing **CAMERA MANUFACTURERS' OPENING MARKMAN BRIEF** was served on counsel of record by email as follows:

For Papst Licensing GmbH & Co. KG (via e-mail):
Robert F. Muse
STEIN, MITCHELL & MEZINES LLP
1100 Connecticut Avenue, NW
Suite 1100
Washington, DC 20036-4195
(202) 737-7777
(202) 296-8312 (fax)
rmuse@steinmitchell.com

Jerold B. Schnayer
HUSCH BLACKWELL SANDERS, WELSH & KATZ
120 South Riverside Plaza
22nd Floor
Chicago, IL 60606
(312) 655-1500
(312) 655-1501 (fax)
jbschnayer@welshkatz.com

For The Victor Company of Japan Parties (via e-mail):
Rachel M. Capoccia, Esq.
HOGAN & HARTSON LLP
1900 Avenue of the Stars
Suite 1400
Los Angeles, California 90067
Email: rmcapoccia@hhlaw.com
Tel: (310) 785-4744
Fax: (310) 785-4601

For The Matsushita Electric Industrial Co., Ltd. Parties (via e-mail):
Richard de Bodo
HOGAN & HARTSON LLP
1900 Avenue of the Stars
Suite 1400
Los Angeles, California 90067
Email: rdebodo@hhlaw.com
Tel: (310) 785-4694
Fax: (310) 785-4601

For The Casio Parties (via e-mail):
J. Kevin Fee
MORGAN, LEWIS & BOCKIUS LLP
1111 Pennsylvania Avenue, NW
Washington, D.C. 20004
jkfee@morganlewis.com

For The Samsung Parties (via e-mail):
Patrick J. Kelleher
Drinker Biddle & Reath LLP
191 North Wacker Drive
Suite 3700
Chicago, IL 60606-1698
Phone: (312) 569-1375
Fax: (312) 569-3375
patrick.kelleher@dbr.com

For the Fujifilm parties. (via e-mail):
Steven J. Routh
Orrick, Herrington & Sutcliffe LLP
Columbia Center
1152 15th St, NW
Washington DC 20005
Phone: (202) 339-8400
Fax: (202) 339-8500
srouth@orrick.com

For the Olympus Parties (via e-mail):
Richard de Bodo
Hogan & Hartson, LLP
1999 Avenue of the Stars
Suite 1400
Los Angeles, CA 90067
Phone: (310) 785-4694
Fax: (310) 785-4601
rdebodo@hhlaw.com

For Ricoh Americas Corp. (via e-mail)
Paul Devinsky
McDermott Will & Emery LLP
600 13th Street N.W.
Washington D.C. 20005-3096
Tel- 202.756.8000 (Main)
Fax -202.756.8087
Mobile-301.512.0060
pdevinsky@mwe.com

For the Nikon Parties (via e-mail)
Marc S. Blackman
Jones Day
77 W. Wacker Drive
Chicago, Illinois 60601
Direct: 312-269-4369
Fax: 312-782-8585
E-mail: msblackman@jonesday.com

For Hewlett-Packard Company (via e-mail):
Heather N. Mewes
Fenwick & West LLP
555 California Street, Suite 1200
San Francisco, CA 94104
Phone: (415) 875-2300
Fax: (415) 281-1350
hmewes@fenwick.com

/s/ Rachel M. Capoccia

Exhibit A

CAMERA MANUFACTURERS' APPENDIX OF
PROPOSED CONSTRUCTIONS OF DISPUTED CLAIM TERMS

The Camera Manufacturers hereby submit the following constructions of terms of the claims in the patents-in-suit that (a) have been identified as requiring construction, and (b) are contained within claims that Papst has asserted against any of the Camera Manufacturers (`399 Patent Claims 1, 2, 3, 5, 7, 11, 14, and 15; `449 Patent Claims 1, 2, 6, 7, 8, 9, 12, 13, 15, 16, 17, and 18). Except as otherwise set forth below, the Camera Manufacturers construe identical (or substantially similar) disputed terms the same in all asserted claims of both patents.

USP 6,470,399's Claim Language Claim 1	Terms Requiring Construction	Proposed Construction
1. An ***interface device for communication between a host device,*** which comprises drivers for input/output devices customary in a host device and a ***multi-purpose interface***, and a ***data transmit/receive device***, the data transmit/receive device being arranged for providing analog data, comprising:	interface device	A stand alone device that can readily physically connect to and disconnect from a host device and a data transmit/receive device and that directs communication between these devices when they are connected.
	host device	A general purpose computer that connects to and controls the operation of peripherals.
	data transmit/receive device	A device that transmits data to and receives data from the host device when connected to the host device by the interface device.
	for communication between [the host device and a data transmit/receive device]	For transmitting of information bidirectionally and actively between two devices.
	multi-purpose interface	A communication interface designed for use with multiple devices having different functions from each other.

USP 6,470,399's Claim Language Claim 1	Terms Requiring Construction	Proposed Construction
a processor;		
a memory;		
a ***first connecting device for interfacing the host device with the interface device*** via the multi-purpose interface of the host device; and	interfacing	Physically connecting
	first connecting device for interfacing the host device with the interface device	A physical plug or connector for permitting a user to readily attach and detach the interface device with the host device.
a ***second connecting device for interfacing the interface device with the data transmit/receive device***, the second connecting device including a sampling circuit for sampling the analog data provided by the data transmit/receive device and an analog-to-digital converter for converting data sampled by the sampling circuit into digital data,	second connecting device for interfacing the interface device with the data transmit/receive device	A physical plug or socket for permitting a user to readily attach and detach the interface device with a plurality of dissimilar data transmit/receive devices.
wherein the interface device is configured by the processor and the memory to include a first command	first command interpreter	A software program for interpreting an inquiry from a host device and sending a signal to the host device in response to the inquiry that "lies to the host computer as to the real nature of the data transmit/receive device."

USP 6,470,399's Claim Language Claim 1	Terms Requiring Construction	Proposed Construction
interpreter and a second command interpreter,	second command interpreter	A software program for translating data request commands from the host into data transfer commands understandable by a plurality of dissimilar data transmit/receive devices.
	wherein the interface device is configured by the processor and memory to include a first command interpreter and a second command interpreter	The processor of the interface device runs a program from its memory to determine the data transfer parameters of the interface device for the first and second command interpreters
wherein the first command interpreter is configured in such a way that the command interpreter, when receiving an ***inquiry*** from the host device as to a type of a device attached to the multi-purpose interface of the host device***, sends a signal, regardless of the type of the data transmit/receive device attached to the second connecting device of the interface device, to the host device which signals to the host device that it is an input/output device customary in a host device***, whereupon the host device communicates with the interface device by means of ***the driver for the input/output device customary in a host device***, and	inquiry	The SCSI inquiry command
	an input/output device customary in a host device	A data input/output device that was normally present within the chassis of most commercially available computers at the time of the invention. The I/O devices that were customary in personal computers were hard disk drives, floppy disk drives, CD-ROM drives and tape drives.
	sends a signal, regardless of the type of the data transmit/receive device attached to the second connecting device of the interface device, to the host device which signals to the host device that it is an input/output device customary in a host device	The first command interpreter sends a signal to the host device that the data transmit/receive device is an input/output device customary in a host device [*circa* 1997], and in so doing "lies to the host computer as to the real nature of the data transmit/receive device." [brackets and italics added]
	the driver	The set of software routines used to control an input/output device.

USP 6,470,399's Claim Language Claim 1	Terms Requiring Construction	Proposed Construction
	the driver for the input/output device customary in a host device	The driver normally present in most commercially available computers at the time of the invention.
wherein the second command interpreter is configured to interpret a data request command from the host device to the type of input/output device signaled by the first command interpreter as a data transfer command for initiating a transfer of ***the digital data*** to the host device.	the digital data	The same digital data output from the analog to digital ("A/D") converter, unmodified by additional processing.

USP 6,470,399's Claim Language Dependent Claims 2, 3, 5, 7	Terms Requiring Construction	Proposed Construction
2. An interface device according to claim 1, wherein the drivers for input/output drivers customary in a host device comprise a hard disk driver, and the signal indicates to the host device that the host device is communicating with a hard disk.	*See* all terms defined with respect to Claim 1	

USP 6,470,399's Claim Language Dependent Claims 2, 3, 5, 7	Terms Requiring Construction	Proposed Construction
3. An interface device according to claim 1, wherein the memory means comprises ***a buffer to buffer data to be transferred between the data transmit/receive device and the host device***.	a buffer to buffer data to be transferred between the data transmit/receive device and the host device	Volatile memory used to temporarily store data to compensate for differences between the rate in the flow of data between the data transmit/receive device and the host device.
5. An interface device according to claim 1, wherein the processor is a digital signal processor.	*See* all terms defined with respect to Claim 1	
7. An interface device according to claim 2, which further comprises a root directory and ***virtual files*** which are present on the signaled hard disk drive and which can be accessed from the host device.	virtual files	A file that does not physically exist as a file in the interface device but appears to the host device to be an actual file, and references data to be transmitted between the data transmit/receive device and the host device.

USP 6,470,399's Claim Language Claim 11	Terms Requiring Construction	Proposed Construction
11. An interface device for communication between a host device, which comprises a multi-purpose interface and a specific driver for this interface, and a data transmit/receive device, the data transmit/receive device being arranged for providing analog data, comprising: a processor; a memory; a first connecting device for interfacing the host device with the interface device via the multi-purpose interface of the host device; and a second connecting device for interfacing the interface device with the data transmit/receive device, the second connecting device including a sampling circuit for sampling the analog data provided by the data transmit/receive device and an analog-to-digital converter for converting data sampled by the sampling circuit into digital data, where the interface device is configured using the processor and the memory to include a first	*See* all terms defined with respect to Claim 1	

USP 6,470,399's Claim Language Claim 11	Terms Requiring Construction	Proposed Construction
command interpreter and a second command interpreter, wherein the first command interpreter is configured in such a way that the interface device, when receiving an inquiry from the host device as to the type of a device attached at the multi-purpose interface of the host device, sends a signal, regardless of the type of the data transmit/receive device attached to the second connecting device of the interface device, to the host device which signals to the host device that it is an input/output device customary in a host device, whereupon the host device communicates with the interface device by means of the ***specific driver for the multi-purpose interface***, and wherein the second command interpreter is configured to interpret a data request command from the host device to the type of input/output device signaled by the first command interpreter as a data transfer command for initiating a transfer of the digital data to the host device.	specific driver for the multi-purpose interface	The set of software routines that control the multi-purpose interface that are developed for the particular multi-purpose interface.

USP 6,470,399's Claim Language Claim 14	Terms Requiring Construction	Proposed Construction
14. A method of communication between a host device, which comprises drivers for input/output devices customary in a host device and a multi-purpose interface, and a data transmit/receive device, the data transmit/receive device being arranged for providing analog data, via an interface device, comprising: interfacing of the host device with a first connecting device of the interface device via the multi-purpose interface of the host device; interfacing of the data transmit/receive device with a second connecting device of the interface device, the second connecting device including a sampling circuit for sampling the analog data provided by the data/transmit/receive device and an analog-to-digital converter for converting data sampled by the sampling circuit into digital data;	*See* all terms defined with respect to Claim 1	
inquiring by the host device at the interface device as to the type of device to which the multi-purpose interface of the host device	inquiring	The SCSI inquiry command

USP 6,470,399's Claim Language Claim 14	Terms Requiring Construction	Proposed Construction
is attached; regardless of the type of the data transmit/receive data attached to the second connecting device of the interface device, responding to the inquiry from the host device by the interface device in such a way that it is an input/output device customary in a host device, whereupon the host device communicates with the interface device by means of the ***usual driver for the input/output device***, and interpreting a data request command from the host device to the type of input/output device customary in the host device as a data transfer command for initiating a transfer of the digital data to the host device.	the usual driver for the input/output device	Indefinite, but to the extent these claim terms can be construed, they should be construed to mean "the driver normally present in most commercially available computers at the time of the invention."

USP 6,470,399's Claim Language Dependent Claim 15	Terms Requiring Construction	Proposed Construction
15. A method according to claim 14, wherein the drivers for input/output devices customary in a host device comprise a driver for a storage device and in particular for a hard disk drive.	*See* all terms defined with respect to Claim 1	

USP 6,895,449's Claim Language Independent Claim 1	Terms Requiring Construction	Proposed Construction
1. An interface device for communication between a host device, which comprises drivers for input/output devices customary in a host device and a multi-purpose interface, and a data transmit/receive device comprising the following features: a processor; a memory; a first connecting device for interfacing the host device with the interface device via the multi-purpose interface of the host device; and a second connecting device for interfacing the interface device with the data transmit/receive device,	*See* all terms defined with respect to `399 Patent	
wherein the interface device is configured by the processor and the memory in such a way that the interface device, when receiving an inquiry from the host device as to the type of a device attached to the multi-purpose interface of the host device, sends a signal, regardless of the type of the data transmit/receive device attached to the second	storage device customary in a host device	Same as "input/output device customary in a host device" of `399, claim 1.

USP 6,895,449's Claim Language Independent Claim 1	Terms Requiring Construction	Proposed Construction
connecting device of the interface device, to the host device which signals to the host device that it is a ***storage device customary in a host device***, whereupon the host device communicates with the interface device by means of ***the driver for the storage device customary in a host device***, and	the driver for the storage device customary in a host device	Same as "the driver for the input/output device customary in a host device" of `399, claim 1.
wherein the interface device is arranged for ***simulating a virtual file system to the host***, the virtual file system including a directory structure.	simulating a virtual file system to the host device	Appearing as a file system to the host, but actually maintaining virtual files, where virtual files do not physically exist as files in the interface device but appear to the host device to be actual files, and reference data to be transmitted between the host device and the data transmit/receive device.

USP 6,895,449's Claim Language Dependent Claims 2, 6, 7, 8, 9, 12, 13, 15, 16	Terms Requiring Construction	Proposed Construction
2. An interface device in accordance with claim 1, in which the directory structure has a configuration file for setting and controlling functions of the interface device or an executable or a batch file for conducting a routine	*See* all terms defined with respect to `399 Patent	

USP 6,895,449's Claim Language Dependent Claims 2, 6, 7, 8, 9, 12, 13, 15, 16	Terms Requiring Construction	Proposed Construction
stored in the memory or a data file used for transferring data from the data transmit/receive device to the host device or a help file for giving help on handling the interface device.		
6. An interface device in accordance with claim 1 wherein, in response to a request from the host to read a boot sequence, the processor is arranged to send a virtual boot sequence to the host.	*See* all terms defined with respect to `399 Patent	
7. An interface device in accordance with claim 6 wherein the virtual boot sequence includes a starting position and a length of a file allocation table, an indication of a type of the storage device or a number of sectors of the storage device.	*See* all terms defined with respect to `399 Patent	
8. An interface device in accordance with claim 7 wherein, in response to a request from the host to display a directory of the storage device, a processor is arranged for transferring the file allocation table and the directory structure to the host.	*See* all terms defined with respect to `399 Patent	
9. An interface device in accordance with claim 1 wherein	*See* all terms defined with respect to `399 Patent	

USP 6,895,449's Claim Language Dependent Claims 2, 6, 7, 8, 9, 12, 13, 15, 16	Terms Requiring Construction	Proposed Construction
the file allocation table and the directory structure is transferred to the host in response to a request from the host to read data from or store data to the storage device.		
12. An interface device in accordance with claim 1 wherein the file allocation table includes information on numbers of blocks occupied by the data file wherein the interface device is arranged for receiving block numbers or a block number range from the host when the host wants to read the data file, and wherein the interface device is arranged to start a data transfer to the host, when the block numbers or the block number range is received from the host.	*See* all terms defined with respect to `399 Patent	
13. An interface device in accordance with claim 12 wherein the processor is arranged for formatting the data acquired by the second connecting device into blocks having a predetermined size, the predetermined size being suited for the storage device.	*See* all terms defined with respect to `399 Patent	
15. An interface device in accordance with claim 1 wherein the storage device is a hard disk.	*See* all terms defined with respect to `399 Patent	

USP 6,895,449's Claim Language Dependent Claims 2, 6, 7, 8, 9, 12, 13, 15, 16	Terms Requiring Construction	Proposed Construction
16. An interface device in accordance with claim 1 wherein the memory has ***a data buffer for permitting independence in terms of time of the data transmit/receive device attachable to the second connecting device from the host device attachable to the first connecting device***.	*See* all terms defined with respect to `399 Patent, Claim 1	
	a data buffer for permitting independence in terms of time of the data transmit/receive device attachable to the second connecting device from the host device attachable to the first connecting device	Volatile memory used to temporarily store data to compensate for differences between the rate in the flow of data between the data transmit/receive device and the host device.

USP 6,895,449's Claim Language Independent Claim 17	Terms Requiring Construction	Proposed Construction
17. An interface device for communication between a host device, which comprises a multi-purpose interface and a specific driver for this interface, and a data transmit/receive device comprising the following features: a processor; a memory; a first connecting device for interfacing the host device with the interface device via the multi-purpose interface of the host device; and	*See* all terms defined with respect to `399 Patent	

USP 6,895,449's Claim Language Independent Claim 17	Terms Requiring Construction	Proposed Construction
a second connecting device for interfacing the interface device with the data transmit/receive device, where the interface device is configured using the processor and the memory in such a way that the interface device, when receiving an inquiry from the host device as to the type of a device attached at the multi-purpose interface of the host device, sends a signal, regardless of the type of the data transmit/receive device attached to the second connecting device of the interface device, to the host device which signals to the host device that it is a storage device customary in a host device, whereupon the host device communicates with the interface device by means of the specific driver for the multi-purpose interface, and wherein the interface device is arranged for simulating a virtual file system to the host, the virtual file system including a file allocation table and a directory structure.		

USP 6,895,449 Claim Language Independent Claim 18	Terms Requiring Construction	Proposed Construction
18. A method of communication between a host device, which comprises drivers for input/output devices customary in a host device and a multi-purpose interface, and a data transmit/receive device via an interface device comprising the following steps: interfacing of the host device with a first connecting device of the interface device via the multi-purpose interface of the host device; interfacing of the data transmit/receive device with a second connecting device of the interface device; inquiring by the host device at the interface device as to the type of device to which the multi-purpose interface of the host device is attached; regardless of the type of the data transmit/receive device attached to the second connecting device of the interface device, responding to the inquiry from the host device by the interface device in such a way that it is a storage device customary in a host device,	*See* all terms defined with respect to `399 Patent	

USP 6,895,449 Claim Language Independent Claim 18	Terms Requiring Construction	Proposed Construction
whereupon the host device communicates with the interface device by means of the usual driver for the storage device, and wherein the interface device is arranged for simulating a virtual file system to the host, the virtual file system including a file allocation table and a directory structure.		

Exhibit B

US005506692A

United States Patent [19]

Murata

[11] Patent Number: 5,506,692

[45] Date of Patent: Apr. 9, 1996

[54] **IMAGE HANDLING APPARATUS HAVING FILE SYSTEM EMULATION MEANS**

[75] Inventor: **Kazuyuki Murata**, Tsuzuki, Japan

[73] Assignee: **Matsushita Electric Industrial Co., Ltd.**, Kadoma, Japan

[21] Appl. No.: **259,530**

[22] Filed: **Jun. 14, 1994**

Related U.S. Application Data

[62] Division of Ser. No. 36,028, Mar. 23, 1993.

[30] **Foreign Application Priority Data**

Apr. 9, 1992 [JP] Japan 4-088541
Aug. 7, 1992 [JP] Japan 4-211102

[51] **Int. Cl.**[6] **H04N 1/00**; H04N 1/21; G06F 3/00; G06F 9/455; G06F 13/00

[52] **U.S. Cl.** **358/442**; 358/444; 395/112; 395/115; 395/500; 395/828; 395/830; 395/882; 395/883; 395/892

[58] **Field of Search** 358/442, 468, 358/444, 296; 395/500, 275, 112, 114, 115, 828, 830, 882, 883, 892

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,088,033 2/1992 Binkley et al. 395/500

OTHER PUBLICATIONS

Operating Systems: Design and Implementation, Andrew S. Tanenbaum, 1987, pp. 299–308.

Primary Examiner—Scott A. Rogers
Attorney, Agent, or Firm—Ratner & Prestia

[57] **ABSTRACT**

An image forming apparatus which is used with an external host computer. The image forming apparatus includes an interface for connecting the image forming apparatus to the external host computer. The image forming apparatus further includes a file system emulator which emulates a file system contained in the disk storage apparatus of the external host computer. The file system emulator makes the image forming apparatus look like the disk storage apparatus to the external host computer connected to the image forming apparatus to transfer images from the external host computer to the image forming apparatus via the interface device.

10 Claims, 7 Drawing Sheets

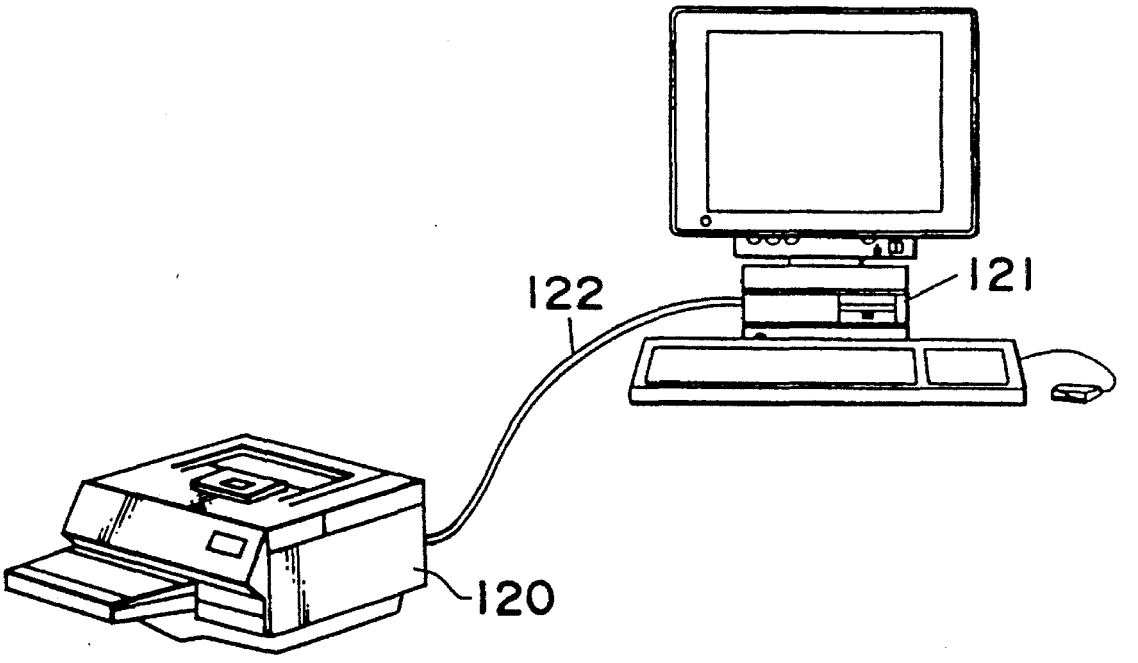


Fig. 1

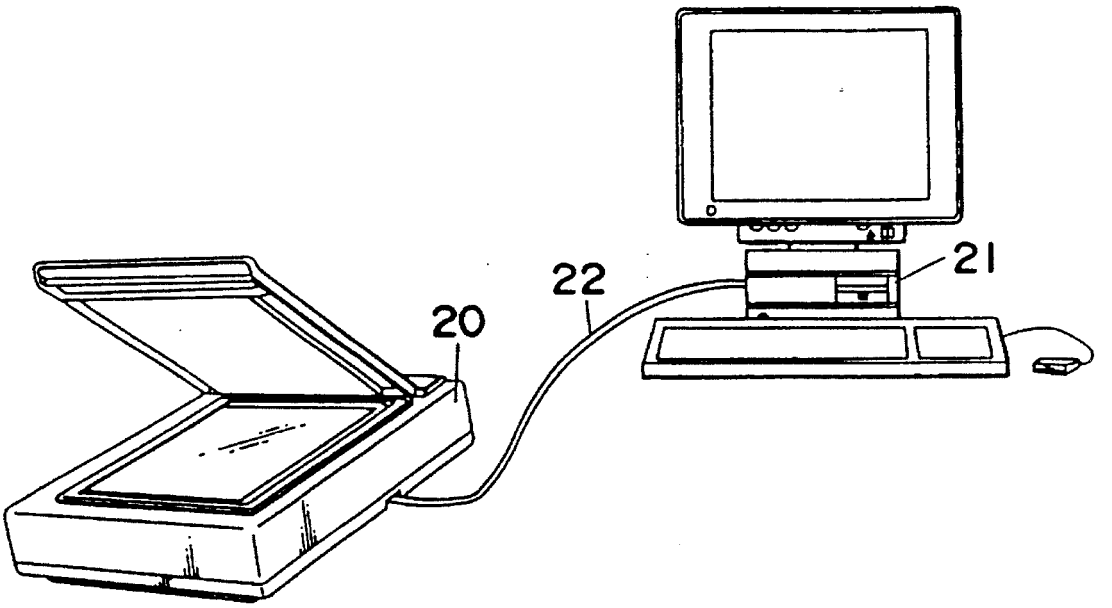
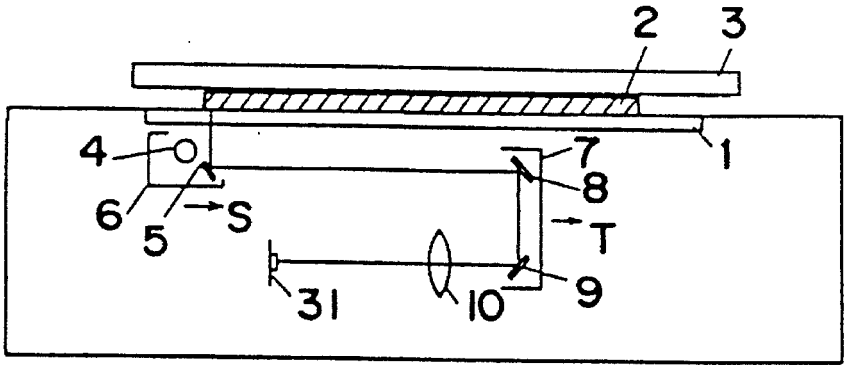


Fig. 2



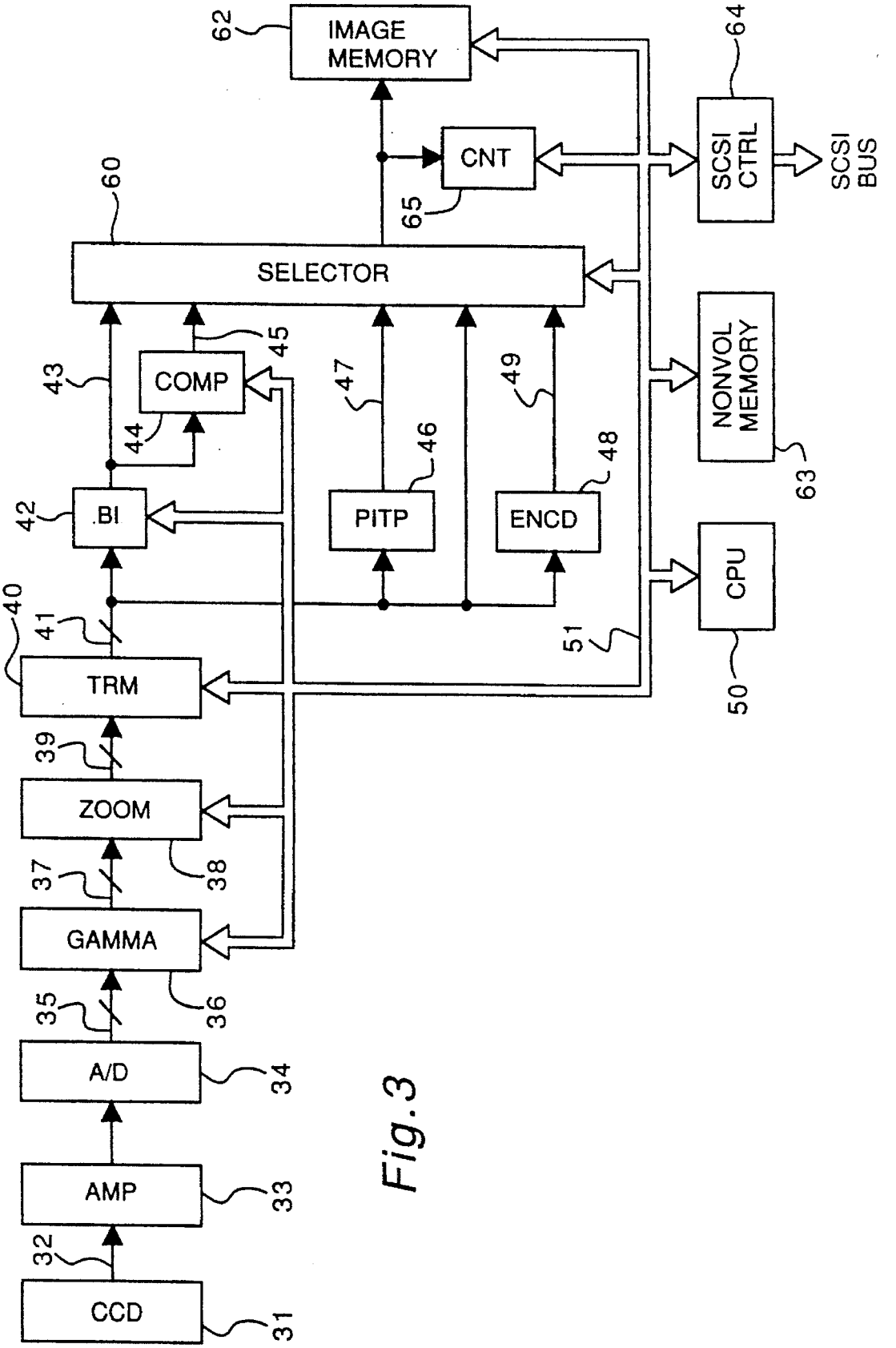

IMAGE MEMORY
62
64
CNT
SCSI CTRL
SCSI BUS
65
60
SELECTOR
NONVOL MEMORY
63
45
43
COMP
47
49
44
46
48
42
BI
PITP
ENCD
CPU
40
41
51
TRM
50
39
ZOOM
37
38
GAMMA
35
36
A/D
34
AMP
33
32
CCD
31

Fig. 3

Fig.4

```
#SCANNER PARAMETER FILE EXAMPLE
# AREA PARAMETER(inch)
# X,Y,XL,YL
1.2
2.4
5.5
6.3
#ZOOMING PARAMETER(%) XZ,YZ
100
150

#IMAGE PROCESSING PARAMETER
#         SIMPLE BI-LEVEL IMAGE                    ->0
#         BI-LEVEL COMPRESSION IMAGE(MH)           ->1
#         BI-LEVEL COMPRESSION IMAGE(MR)           ->2
#         BI-LEVEL COMPRESSION IMAGE(MMR)          ->3
#         ERROR DIFFUSION BI-LEVEL IMAGE           ->4
#         88ITS GRAY SCALE IMAGE                   ->5
#         88ITS. GRAY SCALE COMPRESSION IMAGE      ->6
1

#GAMMA TRANSFER PARAMETER
#         LINEAR              ->0
#         DENSITY             ->1
#         DATE SETTABLE       ->2
2

#GAMMA TRANSFER DATA TABLE
#DATA MUST BE 0 (=.255) =
00 10 17 1C 20 24 27 2A 20 30 32 35 37 3A 3C 3E
40 42 44 46 47 49 4B 4D 4E 50 51 53 54 56 57 59
5A 5C 5D 5E 60 61 62 64 65 66 67 69 6A 6B 6C 6D
6F 70 71 72 73 74 75 76 77 79 7A 7B 7C 7D 7E 7F
80 81 82 83 84 85 86 87 87 88 89 8A 8B 8C 8D 8E
8F 90 91 91 92 93 94 95 96 97 97 98 99 9A 9B 9C
9C 9D 9E 9F A0 A0 A1 A2 A3 A4 A4 A5 A6 A7 A7 A8
A9 AA AA AB AC AD AD AE AF B0 B0 B1 B2 B3 B3 B4
B5 B5 B6 B7 B7 B8 B9 BA BA BB BC BC BD BE BE BF
C0 C0 C1 C2 C2 C3 C4 C4 C5 C6 C6 C7 C7 C8 C9 C9
CA CB CB CC CC CD CE CE CF D0 D0 D1 D1 D2 D3 D3
D4 D4 D5 D6 D6 D7 D7 D8 D9 D9 DA DA DB DC DC DD
DD DE DE DF E0 E0 E1 E1 E2 E2 E3 E4 E4 E5 E5 E6
E6 E7 E7 E8 E9 E9 EA EA EB EB EC EC ED ED EE EE
EF F0 F0 F1 F1 F2 F2 F3 F3 F4 F4 F5 F5 F6 F6 F7
F7 F8 F8 F9 F9 FA FA FB FB FC FC FD FD FE FE FF
```

Fig.5

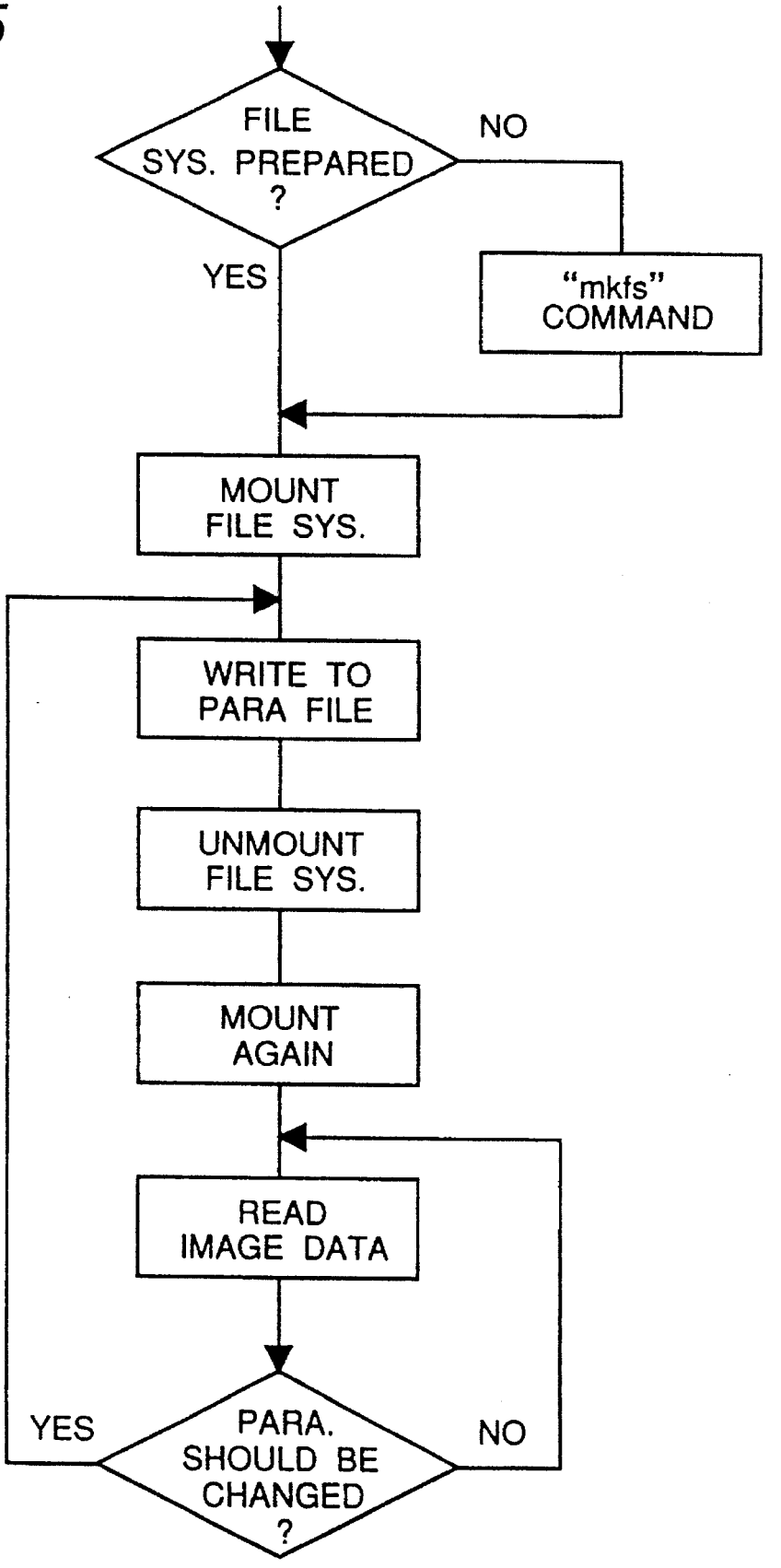

FILE SYS. PREPARED ?
NO
YES
"mkfs" COMMAND
MOUNT FILE SYS.
WRITE TO PARA FILE
UNMOUNT FILE SYS.
MOUNT AGAIN
READ IMAGE DATA
PARA. SHOULD BE CHANGED ?
YES
NO

Fig. 6

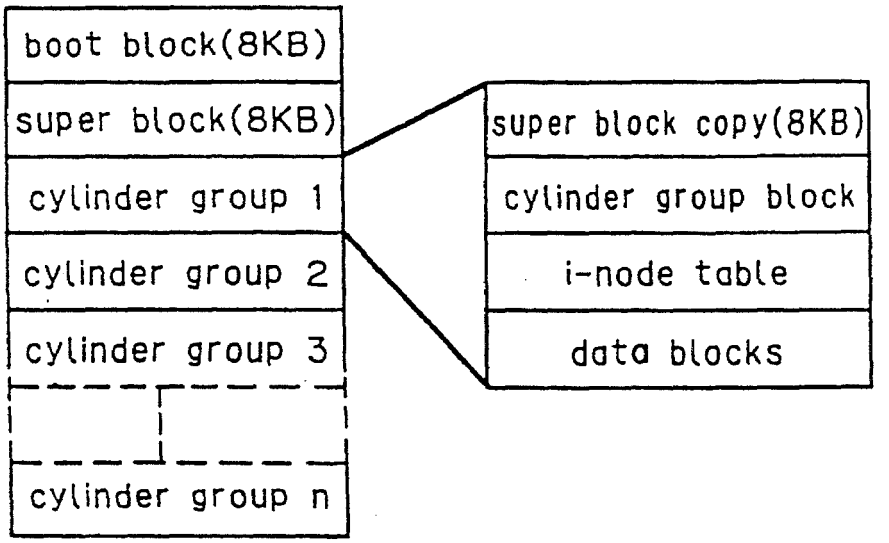

boot block(8KB)
super block(8KB)
cylinder group 1
cylinder group 2
cylinder group 3
cylinder group n
super block copy(8KB)
cylinder group block
i-node table
data blocks

Fig. 7

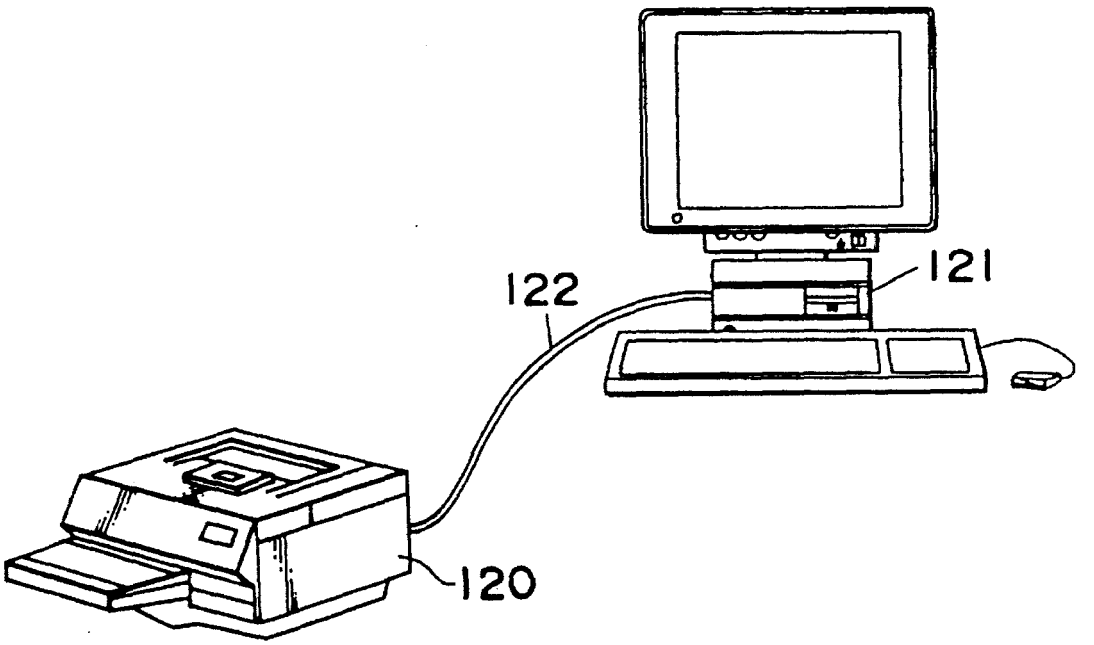

122
121
120

Fig.8

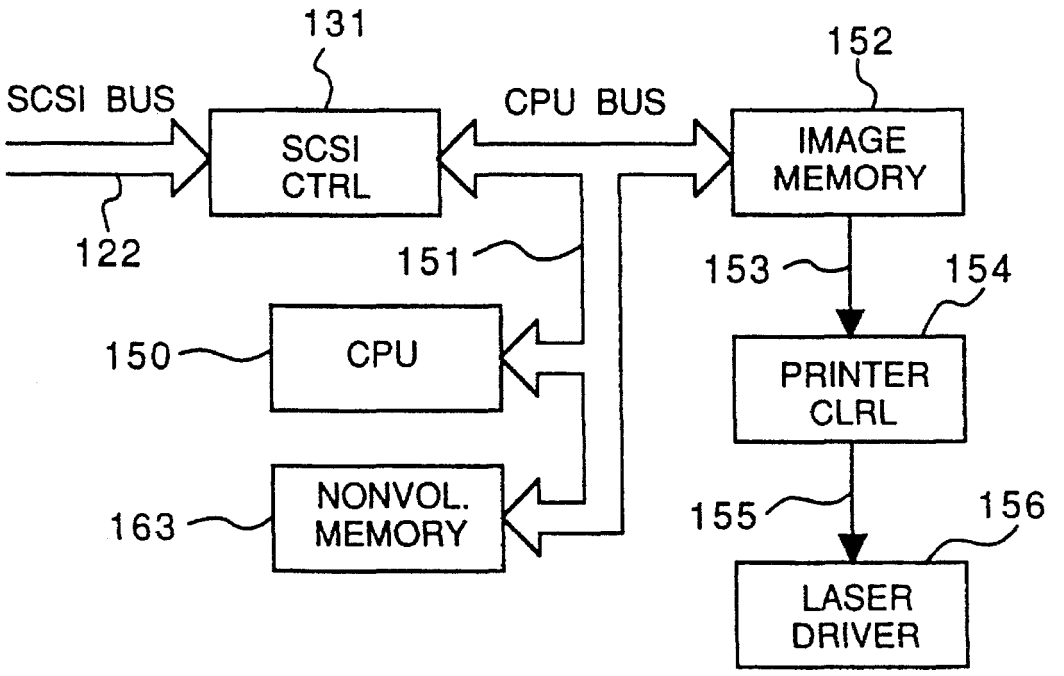


Fig.9

```
#PRINTER PARAMETER FILE EXAMPLE
#PAPER 0->A4 1->B4
0
#PRINTING DIRECTION 0->PORTRAIT 1->LANDSCAPE
0
#PRINT VOLUME
1
```

Fig.10

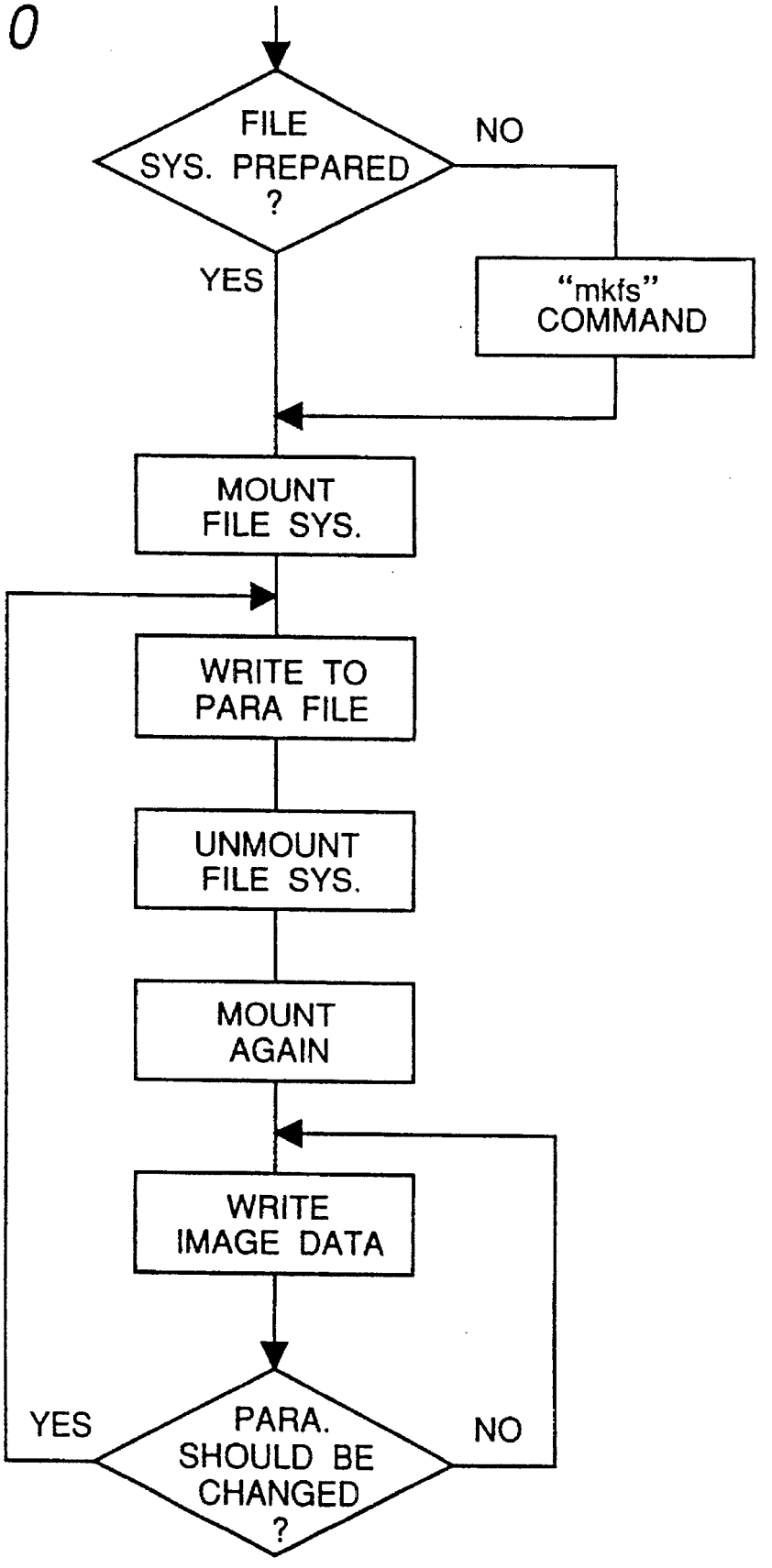

FILE SYS. PREPARED ?
NO
YES
"mkfs" COMMAND
MOUNT FILE SYS.
WRITE TO PARA FILE
UNMOUNT FILE SYS.
MOUNT AGAIN
WRITE IMAGE DATA
PARA. SHOULD BE CHANGED ?
YES
NO

IMAGE HANDLING APPARATUS HAVING FILE SYSTEM EMULATION MEANS

This application is a division of application Ser. No. 08/036,028, filed Mar. 23, 1993, (status: pending)

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates generally to an image handling apparatus, and more particularly to an image scanner and to an image forming apparatus both for transferring data to and from an external host apparatus via an interface.

2. Description of the Prior Art

A computer is generally operatively connected to several peripheral devices such as, for example, a magnetic disc, a magnetic tape, a printer or the like. Recently, a small computer system interface (SCSI) is standardized as an interface means for carrying out high-speed data transfer. Through the standardization, the SCSI is in wide practical use today as an interface for various computers.

By virtue of marked improvement in performance of small computers, e.g. workstations, the development from character codes to bit-map data, which has hitherto been carried out in a printer, tends to be carried out in the computer using outline font data which the computer has. Whereas the technique of developing, for example, fonts to the bit-map data in the computer has advantages in adding other fonts, the use of the SCSI is inevitably required for high-speed data transfer because the quantity of data to be transferred from the computer to the printer is increased.

In applications where an image scanner or an image forming apparatus is connected to an SCSI of a computer employed as a host computer, and parameter setting for such apparatus or image data transfer is carried out by the computer, the computer is required to have a software "device driver" for the apparatus connected thereto.

Because image scanners or image forming apparatus are not standardized in kind of parameters which can be set or in functions, the device driver therefor is not generally contained in an operating system (OS) of the computer. Accordingly, it is necessary to prepare the device driver for the image scanner or the image forming apparatus connected to the host computer.

Under the present conditions discussed above, however, in order to enable a certain image scanner or image forming apparatus to be connected to any one of various types of host computers, it is necessary to prepare a device driver for each type of host computer. As a result, the problem arises that the preparation of the device driver requires much labor and increases costs.

SUMMARY OF THE INVENTION

The present invention has been developed to overcome the above-described disadvantages.

It is accordingly an object of the present invention to provide an improved image handling apparatus, for example an image scanner or an image forming apparatus, which requires no preparation of any new device driver.

In accomplishing the above and other objects, an image scanner according to the present invention comprises means for reading an image, an interface means for connecting the image scanner to an external host apparatus, and a file system emulation means for emulating a file system contained in the external host computer.

When the present invention is applied to an image forming apparatus, the read means is replaced with an image forming means for forming an image on a recording medium.

Because an operating system of a computer constructs a file system in a hard disc, there invariably exists a device driver for the hard disc.

As discussed hereinabove, because the image scanner or image forming apparatus according to the present invention is provided with the file system emulation means, the control of the apparatus or the transfer of image data can be carried out using the device driver for existing hard discs. Furthermore, because the operating system is provided with various commands or system calls which are utilized to access to the file system, development of application software for use in a host computer operatively connected to the image scanner or image forming apparatus is facilitated.

In applications where the image scanner or image forming apparatus according to the present invention is connected to an external host computer, it is not necessary to prepare the device driver for each type of computer if the file system of the computer is the same. In short, the apparatus can be connected to any one of various types of computers having the same file system, e.g. any one of all computers having software called the "UNIX" as an operating system.

BRIEF DESCRIPTION OF THE DRAWINGS

The above and other objects and features of the present invention will become more apparent from the following description of preferred embodiments thereof with reference to the accompanying drawings, throughout which like parts are designated by like reference numerals, and wherein:

FIG. **1** is a perspective view of an image scanner according to the present invention which is connected to an external host computer;

FIG. **2** is a schematic sectional view of the image scanner of FIG. **1**;

FIG. **3** is a block diagram of the image scanner of FIG. **1**;

FIG. **4** is a programmed file that is read by the computer when parameters are set in;

FIG. **5** is a flowchart indicating the procedure at the time the image scanner is controlled by a workstation;

FIG. **6** is a schematic view indicating the layout of a file system contained in the workstation;

FIG. **7** is a perspective view of an image forming apparatus according to the present invention which is connected to an external host computer;

FIG. **8** is a block diagram of the image forming apparatus of FIG. **7**;

FIG. **9** is a programmed file that is read by the computer when parameters are set in; and

FIG. **10** is a flowchart indicating the procedure at the time the image forming apparatus is controlled by a workstation.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Referring now to the drawings, there is shown in FIG. **1** an image scanner **20** embodying the present invention. The image scanner **20** is connected to an external host computer via an SCSI bus **22**. In FIG. **1**, the external host computer is a workstation **21** having the "UNIX" as an operating system. A hard disc in which a file system for the workstation **21** has been formulated is accommodated in the workstation **21** and

is connected to the SCSI bus **22** inside the workstation **21**.

FIG. **2** schematically depicts the internal construction of the image scanner **20** according to the present invention. The image scanner **20** comprises a document platform **1** made of glass on which a document **2** is to be placed with the image surface thereof directed downwards, a document cover **3** to be overlaid on the document **2** to hold the document **2**, and a scanning unit **6** comprising a fluorescent lamp **4** and a reflection mirror **5**. The scanning unit **6** is driven by a motor (not shown) and is moved in a direction shown by an arrow S at a constant speed to carry out sub-scanning with respect to the document **2**. The image scanner **20** further comprises a half-speed unit **7**, a lens **10**, and a line-type CCD image sensor **31**. The half-speed unit **7** comprises two mirrors **8** and **9**. During scanning, when the scanning unit **6** is moved in the direction of the arrow S, the half-speed unit **7** is moved in a direction shown by an arrow T at a speed half of the scanning unit **6**. Reflected light from the document **2** is further reflected by the mirrors **5**, **8**, and **9**, and is focused on the CCD image sensor **31** by the lens **10** for image formation. The CCD image sensor **31** carries out main-scanning in the line direction with respect to the reflected light from the document **2** to convert it to an electric signal.

As shown in FIG. **3**, the CCD **31** reads the reflected light from the document **2** at a resolution of 400 dpi, converts it to the electric signal, and outputs an analogue image signal **32**. The analogue image signal **32** is then amplified by an amplifier **33** and is converted to a digital image signal **35** by an 8-bit A/D converter **34**. A gamma transfer circuit **36** carries out digital-to-digital conversion for conversion of the gradation characteristic, thereby converting the digital image signal **35** to an image signal **37**. The gamma transfer circuit **36** is comprised of a look-up table formulated by the use of a RAM. The RAM is connected to a CPU bus **51** of a CPU **50**, and data stored therein can be set by the CPU **50**. Accordingly, the conversion characteristic of the gamma transfer circuit **36** can be changed by the CPU **50**.

A zooming circuit **38** carries out a zooming operation in the direction of main-scanning by interpolating or thinning out the image signal **37**, and outputs an image signal **39**. The CPU **50** sets the zooming rate of the zooming circuit **38** via the CPU bus **51**. The zooming operation in the sub-scanning direction is carried out by changing the speed of movement of the scanning unit **6** shown in FIG. **2**. A trimming circuit **40** carries out a trimming operation wherein only part of the image signal **39** that is indicative of a predetermined rectangular region on the document **2** is made effective, and outputs an image signal **41**. The trimming circuit **40** is connected to the CPU bus **51** of the CPU **50**, and the trimming region can be set by the CPU **50**.

A binary circuit **42** compares the image signal **41** with a predetermined threshold value and outputs a binary image signal **43**. The threshold value can be set by the CPU **50**. A compression circuit **44** encodes and compresses the binary image signal **43** and outputs a compressed binary image signal **45**. The compression circuit **44** carries out the encoding using any one of three binary image encoding methods: MH; MR; and MMR. The selection of an appropriate compression method is carried out by the CPU **50**. A pseudo-half-tone processor **46** processes the image signal **41** using the error diffusion method and outputs an image signal **47**. An encoder **48** encodes and compresses the image signal **41** and outputs an encoded image signal **49**.

A selector **60** selects an appropriate signal from among the compressed binary image signal **45**, the binary image signal **43**, the image signal **47**, the image signal **41**, and the encoded image signal **49**. After the selection, the selector **60** outputs an image signal **61**. The appropriate signal selection is carried out by the CPU **50** via the CPU bus **51**. An image memory **62** stores the image signal **61** selected by the selector **60**. Because the image memory **62** is connected to the CPU bus **51**, the image memory **62** is freely accessible from the CPU **50** and an SCSI controller **64**. A data counter **65** counts the quantity of image data to be read and that of image date actually stored in the image memory **62**. Individual values counted can be read from the CPU **50** via the CPU bus **51**.

Data stored in a nonvolatile memory **63** can be read or rewritten by the CPU **50** via the CPU bus **51**. The SCSI controller **64** is a controller, controlled by the CPU **50**, for carrying out data transfer to and from the external host computer via the SCSI.

The operation of the workstation **21** and the image scanner **20** at the time the image scanner **20** is controlled by the workstation **21** is discussed hereinafter.

The image scanner **20** emulates the file system of the "UNIX" as if it were a hard disc. Accordingly, the image scanner **20** looks like the hard disc from the workstation **21** and can be handled as the hard disc.

The image scanner **20** according to the present invention is controlled by the workstation **21** as follows.

When the image scanner **20** connected to the workstation **21** is operated by the workstation **21** for the first time, the workstation **21** prepares a file system in the image scanner **20**, as is the case with the hard disc. In practice, an "mkfs" or "newfs" command of the "UNIX" is executed. At this moment, the operating system provides a device file and a device driver required for operating the hard disc as those required for preparing the file system. The preparation of the file system enables basic information of the file system to be written to a predetermined region, i.e. a super block of the hard disc (image scanner).

In the image scanner **20**, the information written to the super block is stored in the nonvolatile memory **63**. Thereafter, when a request to read the data stored in the super block is sent from the workstation **21**, the image scanner **20** outputs the data stored in the nonvolatile memory **63**. Because the information of the super block is stored in the nonvolatile memory **63**, even if a power supply to the image scanner **20** is cut off, further preparation of the file system is no longer required.

As discussed above, if the preparation of the file system is carried out with respect to the image scanner **20**, the workstation **21** can "mount" the image scanner **20** which emulates the file system, as is the case with the file system of the hard disc. The "mounting" operation is to establish linkage between the file system and device files in the hard disc. After the "mounting" operation, the workstation **21** can access to files contained in the file system prepared in the hard disc. The "mounting" operation is executed using a "mount" command of the "UNIX".

Upon completion of the "mounting" operation, when the file system emulated by the image scanner **20** is viewed from the workstation **21**, one file seems to exist therein. This file is a parameter file required to set parameters for the image scanner **20**. In order for the image scanner **20** to pretend to contain the parameter file therein, as viewed from the workstation **21**, i-node data having information required for the workstation **21** to access to the parameter file are created in the image scanner **20** and are transferred to the workstation **21** according to a request from the workstation **21**.

The name of the parameter file is "scan. para". The attribute of the parameter file is "write only". The parameter

file is a file required to set the operation mode of the image scanner **20** such as, for example, the zooming ratio, the read area, the gamma transfer characteristic, the binary process or the like, and is represented using a predetermined format.

FIG. **4** depicts one example of the parameter file. Lines starting from "#" are comment lines. The parameters are represented in the following order using numerals.

(1) Read area: upper left coordinates (X, Y); length in the direction of X (XL); and length in the direction of Y (YL), (unit: inch)

(2) Zoom. ratio: zooming ratio in the direction of X (XZ); and zooming ratio in the direction of Y (YZ), (unit: %)

(3) Image process. method: binary process or the presence or absence of the compression process

(4) Gamma trans. mode: no transfer when 1; transfer from reflectance to density when 2; and designation of gamma transfer table when 3

(5) γ trans. table: to be represented by hexadecimal numbers

The workstation **21** sets the operation mode of the image scanner **20** by writing the parameter file to the image scanner **20**. The writing of the parameter file for the setting of the operation mode of the image scanner **20** can greatly facilitate the setting of the parameters of the image scanner **20**.

Because the "UNIX" operating system executes buffering such that data to be read from or written to the file system are temporarily stored in a buffer of a main memory of the workstation, it is necessary to "unmount" the file system in order to actually write the data to the hard disc. A "umount" command of the "UNIX" is used for this purpose.

The image scanner **20** reads a document in accordance with the parameters at the time the parameter file has been written. At this moment, the read image data are stored in the image memory **62** inside the image scanner **20**. When the quantity of the image data is greater than the capacity of the image memory **62**, the image data as many as the image memory **62** can accommodate are temporarily stored in the image memory **62**.

Subsequently, in order for the workstation **21** to read the image data from the image scanner **20**, the file system to be emulated by the image scanner **20** is "mounted" again. As viewed from the workstation **21**, there seem to exist two files in the file system emulated by the image scanner **20**. One of them is the aforementioned parameter file "scan para" whereas the other is an image data file, the name of which is "image. data". The attribute of the image data file is "read only".

In order for the image scanner **20** to pretend to contain the image data file therein, as viewed from the workstation **21**, the CPU **50** of the image scanner **20** creates i-node data having information required for the workstation **21** to access to the image data file and transfers them to the workstation **21** in response to a request from the workstation **21**. The size of the image data file as viewed from the workstation **21** depends upon the operation mode which has been set in the image scanner **20** by writing to the parameter file. The workstation **21** reads the contents of this data file so that the image data read by the image scanner **20** may be transferred to the workstation **21**.

FIG. **5** depicts a flowchart indicating the above-described control procedure at the time the image scanner **20** according to the present invention is controlled by the workstation **21**.

The operation of the image scanner **20** is discussed hereinafter in association with the procedure of the workstation **21**.

FIG. **6** schematically depicts an ordinary layout of the file system of the workstation containing therein the "UNIX" as the OS. First 8 KB of the file system is allocated to a boot block including boot programs required for booting the workstation **21**. The next 8 KB is an area called the super block which is generally used to store the then conditions of the file system such as, for example, the number of files, the size of the file system and the like. The super block is followed by a plurality of cylinder groups. Each cylinder group is made up of a copy of the super block, a cylinder group block, an i-node table, and data blocks. Data such as the number of i-nodes, that of the data blocks, an i-node map used, or a map of empty blocks are stored in the cylinder data blocks. The i-node is information indicating the attribute of a file or the location of a data block in which the file exists. Because a directory is also handled as a file, i-nodes are required by the number of files and directories. A predetermined extent is allocated to the area of the i-node table including a plurality of i-nodes at the time the file system is first prepared.

The image scanner **20** which emulates the file system having the "UNIX" and shown in FIG. **6** operates as follows.

The operation at the time the workstation **21** prepares the file system is initially discussed. The file system used herein is a file system to be emulated by the image scanner **20**. The workstation **21** writes to the boot block and to the super block positioned at the first 8 KB and at the next 8 KB of the file system, respectively. The CPU **50** of the image scanner **20** stores these 16 KB data in the nonvolatile memory **63**. When a data read request for reading the data of these blocks is sent from the workstation **21**, the CPU **50** reads the data from the nonvolatile memory **63** and transfers them to the workstation **21**. Likewise, when a data write request for writing data into these blocks is sent from the workstation **21**, the CPU **50** writes the data to the nonvolatile memory **63**.

Thereafter, the workstation **21** writes a copy data of the super block to the first 8 KB of each cylinder block of the file system. At this moment, the CPU **50** can know from the data of the super block, the location of each cylinder in the file system. Accordingly, when the CPU **50** has received the copy data of the super block of the file system from the workstation **21**, the CPU **50** discards such data. On the other hand, when a read request is received, the data of the super block are outputted from the nonvolatile memory **63**. Alternatively, the data of each copy of the super block may be stored in the nonvolatile memory **63**. In this case, when a read request is received, the copy data of the super block are outputted from the nonvolatile memory **63**.

Thereafter, the workstation **21** writes to the cylinder group block of each cylinder group. The CPU **50** can know the location and size of the cylinder group block in the file system from the data of the super block. The CPU **50** stores the data of the cylinder group block in the nonvolatile memory **63**. When a read request of such data is sent from the workstation **21**, the CPU **50** reads the data from the nonvolatile memory **63** and transfers them to the workstation **21**. Likewise, when a data write request is sent from the workstation **21**, the CPU **50** writes data to the nonvolatile memory **63**.

Subsequently, the workstation **21** carries out initialization with respect to the i-node table of each cylinder group. The CPU **50** can know the location and size of the i-node table in the file system from the data of the cylinder group block. The CPU **50** discards data to be written to the i-node table.

When the workstation **21** "mounts" the file system emulated by the image scanner **20**, the workstation **21** reads the

super block, the cylinder group block, and the i-node table from the file system. When the CPU **50** has received a read request for reading the super block and the cylinder group block, the CPU **50** reads the corresponding data from the nonvolatile memory **63** and transfers them to the workstation **21**. The CPU **50** creates a root directory and i-node data of the parameter file "scan. para" and transfers them to the workstation **21**.

The CPU **50** knows the location of the data region of the parameter file in the file system from the created i-node data of the parameter file. Accordingly, the CPU **50** can know the writing of the parameter data from the workstation **21** to the parameter file "scan. para". The CPU **50** makes an analysis of the data written to the parameter file, obtains the parameters, and reads the document in accordance with the set operation mode.

The read image data are stored, from the beginning of the image, in the image memory **62** in accordance with a read region set in the parameter file. When the capacity of the image memory **62** is less than the quantity of the image data, a limited quantity of image data corresponding to the capacity of the image memory **62** is stored. The CPU **50** can know from the value of the data counter **65**, the quantity of the whole image data to be read, that of the image data stored in the image memory **62**, and the region of the image stored in the image memory **62**.

After the image reading, when a read request of the i-node table of the file system is sent from the workstation **21**, the CPU **50** creates the root directory, the i-node of the parameter file "scan. para", and that of the image data file "image. data", and transfers them to the workstation **21**. The size of the image data file is equal to that of the whole image data detected by the data counter **65**.

When a read request for reading data located at a certain position in the image data file is sent from the workstation **21**, and if the requested data exist in the image memory **62**, the CPU **50** transfers such data to the workstation **21** via the SCSI controller **64**. If the image memory **62** contains no requested data, the CPU **50** reads a region of the document corresponding to the requested data and stores it in the image memory **62**. Thereafter, the CPU **50** takes out the data from the image memory **62** and transfers them to the workstation **21**. As described above, because the image scanner **20** has an image memory, it can take full advantage of high-speed data transfer through the SCSI. Furthermore, because the image memory is not necessarily required to have a capacity sufficient to store the entire region to be read, the memory need not have a capacity unduly greater than that necessary.

It is to be noted here that a magnetic storage may be used in place of the nonvolatile memory **63**.

FIG. **7** depicts an image forming apparatus **120** according to the present invention, which is connected to an external host computer via an SCSI bus **122**. The external host computer shown in FIG. **7** is a workstation **121** having the "UNIX" as an operating system. A hard disc in which a file system for the workstation **121** has been formulated is accommodated in the workstation **121** and is connected to the SCSI bus **122** inside the workstation **121**.

FIG. **8** is a block diagram of the image forming apparatus **120** according to the present invention.

Data stored in a nonvolatile memory **163** can be read or rewritten by a CPU **150** via a CPU bus **151**. An SCSI controller **164** carries out data transfer to and from the external host computer via the SCSI and is controlled by the CPU **150**.

An image memory **152** stores bit-map image data transferred via the SCSI bus **122** by at least one page. At the time the image data have been stored by one page in the image memory **152**, a printer engine (not shown) commences an image forming operation, and image data **153** are outputted from the image memory **152** to a printer engine controller **154** in synchronism with the operation of an image forming unit. The printer engine controller **154** outputs a laser modulation signal **155** to a laser driver **156** based on the image signal **153**. Because the image forming unit comprises a conventionally known laser printer employing a semiconductor laser, explanation thereof is omitted.

The operation of the workstation **121** and the image forming apparatus **120** at the time the image forming apparatus **120** is controlled by the workstation **121** is discussed hereinafter.

The image forming apparatus **120** emulates a file system of the "UNIX" as if it were a hard disc. Accordingly, the image forming apparatus **120** looks like the hard disc from the workstation **121** and can be handled as the hard disc.

The image forming apparatus **120** according to the present invention is controlled by the workstation **121** as follows.

When the image forming apparatus **120** connected to the workstation **121** is operated by the workstation **121** for the first time, the workstation **121** prepares a file system in the image forming apparatus **120**, as is the case with the hard disc. In practice, an "mkfs" or "newfs" command of the "UNIX" is executed. At this moment, the operating system provides a device file and a device driver required for operating the hard disc as those required for preparing the file system. The preparation of the file system enables basic information of the file system to be written to a predetermined region, i.e. a super block of the hard disc.

In the image forming apparatus **120**, the information written to the super block is stored in the nonvolatile memory **163**. Thereafter, when a request to read the data stored in the super block is sent from the workstation **121**, the image forming apparatus **120** outputs the data stored in the nonvolatile memory **163** to the SCSI bus **122**. Because the information of the super block is stored in the. nonvolatile memory **163**, even if a power supply to the image forming apparatus **120** is cut off, further preparation of the file system is no longer required.

As discussed above, if the preparation of the file system is carried out with respect to the image forming apparatus **120**, the workstation **121** can "mount" the image forming apparatus **120** which emulates the file system, as is the case with the file System of the hard disc.

Upon completion of the "mounting" operation, when the file system emulated by the image forming apparatus **120** is viewed from the workstation **121**, one file seems to exist therein. This file is a parameter file required to set parameters for the image forming apparatus **120**. In order for the image forming apparatus **120** to pretend to contain the parameter file therein, as viewed from the workstation **121**, i-node data having information required for the workstation **121** to access to the parameter file are created in the image forming apparatus **120** and are transferred to the workstation **121** according to a request from the workstation **121**.

The name of the parameter file is "printer. para". The attribute of the parameter file is "write only". The parameter file is a file required to set the operation mode of the image forming apparatus **120** such as, for example, the direction of printing (portrait or landscape), the size of recording papers (A4, B4 etc.), the number of prints, or the like, and is represented using a predetermined format.

FIG. **9** depicts one example of the parameter file. Lines starting from "#" are comment lines. The parameters are represented in the following order using numerals.

(1) Size of recording papers to be used

(2) Direction of printing

(3) Number of prints

The workstation **121** sets the operation mode of the image forming apparatus **120** by writing the parameter file to the image forming apparatus **120**. The writing of the parameter file for the setting of the operation mode of the image forming apparatus **120** can greatly facilitate the setting of the parameters of the image forming apparatus **120**.

Because the "UNIX" operating system executes buffering such that data to be read from or written to the file system are temporarily stored in a buffer of a main storage, it is necessary to "unmount" the file system in order to actually write the data to the hard disc.

The image forming apparatus **120** sets the operation mode thereof in accordance with the parameters at the time the parameter file has been written.

Thereafter, in order for the workstation **121** to write the image data to the image forming apparatus **120**, the file system to be emulated by the image forming apparatus **120** is "mounted" again. As viewed from the workstation **121**, there seem to exist two files in the file system emulated by the image forming apparatus **120**. One of them is the aforementioned parameter file "printer. para", whereas the other is an image data file, the name of which is "image. data". The attribute of the image data file is "write only".

In the case where the resolution of the image forming apparatus **120** is chosen to be 300 dpi (84.667 μm/pixel), and the size of recording papers and the direction of printing are A4 and portrait, respectively, the size V of the image data file is given by:

$$V=\text{int}((\text{int}(210/0.084667+0.5)+7)/8)\times\text{int}(297/0.084+0.5)$$
$$=1087480 \text{ (bites)}$$

where "int" is a symbol to round down a numeral in parentheses.

In order for the image forming apparatus **120** to pretend to contain the image data file therein, as viewed from the workstation **21**, the CPU **150** of the image forming apparatus **120** creates i-node data having information required for the workstation **121** to access to the image data file and transfers them to the workstation **121** in response to a request from the workstation **121**. The size of the image data file as viewed from the workstation **121** depends upon the operation mode which has been set in the image forming apparatus **120** by writing to the parameter file. The workstation **121** writes the image data to the image data file so that the image data may be transferred from the workstation **121** to the image forming apparatus **120**.

FIG. **10** depicts a flowchart indicating the above-described control procedure at the time the image forming apparatus **120** is controlled by the workstation **121**.

The operation of the image forming apparatus **120** is discussed hereinafter in association with the procedure of the workstation **121** with reference to the file system of the "UNIX" shown in FIG. **6**.

The operation at the time the workstation **121** prepares the file system is initially discussed. The file system discussed hereinbelow is a file system to be emulated by the image forming apparatus **120**. The workstation **121** writes to the boot block and to the super block positioned at the first 8 KB and at the next 8 KB of the file system, respectively. The CPU **150** of the image forming apparatus **120** stores these 16 KB data in the nonvolatile memory **163**. When a data read request for reading the data of these blocks is sent from the workstation **121**, the CPU **150** reads the data from the nonvolatile memory **163** and transfers them to the workstation **121**. Likewise, when a data write request for writing data into these blocks is sent from the workstation **121**, the CPU **150** writes the data to the nonvolatile memory **163**.

Thereafter, the workstation **121** writes a copy data of the super block to the first 8 KB of each cylinder block of the file system. At this moment, the CPU **150** can know from the data of the super block, the location of each cylinder in the file system. Accordingly, when the CPU **150** has received the copy data of the super block of the file system from the workstation **121**, the CPU **150** discards such data. On the other hand, when a read request is received, the data of the super block are outputted from the nonvolatile memory **163**. Alternatively, the data of each copy of the super block may be stored in the nonvolatile memory **163**. In this case, when a read request is received, the copy data of the super block are outputted from the nonvolatile memory **163**.

Thereafter, the workstation **121** writes to the cylinder group block of each cylinder group. The CPU **150** can know the location and size of the cylinder group block in the file system from the data of the super block. The CPU **150** stores the data of the cylinder group block in the nonvolatile memory **163**. When a read request of such data is sent from the workstation **121**, the CPU **150** reads the data from the nonvolatile memory **163** and transfers them to the workstation **121**. Likewise, when a data write request is sent from the workstation **121**, the CPU **150** writes data to the nonvolatile memory **163**.

Subsequently, the workstation **21** carries out initialization with respect to the i-node table of each cylinder group. The CPU **50** can know,the location and size of the i-node table in the file system from the data of the cylinder group block. The CPU **50** discards data to be written to the i-node table.

When the workstation **121** "mounts" the file system emulated by the image forming apparatus **120**, the workstation **121** reads the super block, the cylinder group block, and the i-node table from the file system. When the CPU **150** has received a read request for reading the super block and the cylinder group block, the CPU **150** reads the corresponding data from the nonvolatile memory **163** and transfers them to the workstation **121**. The CPU **150** creates a root directory and i-node data of the parameter file "printer. para" and transfers them to the workstation **121**.

The CPU **150** knows the location of the data region of the parameter file in the file system from the created i-node data of the parameter file. Accordingly, the CPU **150** can know the writing of the parameter data from the workstation **121** to the parameter file "printer. para". The CPU **150** makes an analysis of the data written to the parameter file, obtains the parameters, and carries out an image forming operation in accordance with the set operation mode.

When a read request of the i-node table of the file system is sent from the workstation **121**, the CPU **150** creates the root directory, the i-node of the parameter file "printer. para", and that of the image data file "image. data", and transfers them to the workstation **121**. The size of the image data file is determined by the parameters set in the parameter file.

When a data write request for writing data to a certain position in the image data file is sent from the workstation **121**, the CPU **150** calculates the address on the image memory from the location on the image file and transfers the image data to the image memory **152** via the SCSI controller **131**. Upon completion of the writing of the last one bite of the image data file, the CPU **150** detects that the transfer of the image data has been completed by one page and outputs the image data from the image memory **152** to the printer engine controller **154**.

It is to be noted here that although the nonvolatile memory **163** is employed in the image forming apparatus **120**, a magnetic storage may be used in place thereof.

As discussed hereinabove, because the image scanner or image forming apparatus according to the present invention is provided with a file system emulation means for emulating a file system, the control of the apparatus or the transfer of image data can be carried out using a device driver for existing hard discs. Furthermore, because the operating system is provided with various commands or system calls which are utilized to access to the file system, development of application software for use in a host computer operatively connected to the image scanner or image forming apparatus is facilitated.

In applications where the image scanner or image forming apparatus according to the present invention is connected to an external host computer, it is not necessary to prepare the device driver for each type of computer if the file system of the computer is the same. In short, the apparatus can be connected to any one of various types of computers having the same file system.

If the image scanner is a full color scanner, or if the image forming apparatus is a full color printer, it is necessary to transfer a large quantity of data, for example 24 bits per 1 pixel, from the external host computer. Even in this case, high-speed data transfer can be carried out using the SCSI.

In addition, the combined use of both of the image scanner and the image forming apparatus according to the present invention provides a copier wherein a copying operation is carried out by copying image data files of the two apparatus.

Although the present invention has been fully described by way of examples with reference to the accompanying drawings, it is to be noted here that various changes and modifications will be apparent to those skilled in the art. Therefore, unless such changes and modifications otherwise depart from the spirit and scope of the present invention, they should be construed as being included therein.

What is claimed is:

1. An image forming apparatus for use with an external host apparatus containing a disk storage apparatus, said image forming apparatus comprising:

means for forming an image;

an interface means for connecting the image forming apparatus to the external host apparatus; and

a file system emulation means for:

(a) emulating a file system contained in the disk storage apparatus of the external host apparatus, and

(b) making the image forming apparatus look like the disk storage apparatus to the external host apparatus connected to the image forming apparatus to transfer said image from said external host apparatus to said image forming means via said interface means,

said file system emulation means comprising a memory means for storing data of a predetermined region of said file system, to which the region basic information of the file system has been written.

2. An image forming apparatus for use with an external host apparatus containing a disk storage apparatus, said image forming apparatus comprising:

means for forming an image;

an interface means for connecting the image forming apparatus to the external host apparatus; and

a file system emulation means for:

(a) emulating a file system contained in the disk storage apparatus of the external host apparatus, and

(b) making the image forming apparatus look like the disk storage apparatus to the external host apparatus connected to the image forming apparatus to transfer said image from said external host apparatus to said image forming means via said interface means,

said file system emulation means comprising a parameter setting means for creating a parameter file required to set an operation mode of the image forming apparatus based on parameter data written to the parameter file from the external host apparatus.

3. An image forming apparatus for use with an external host apparatus containing a disk storage apparatus, said image forming apparatus comprising:

an image forming means for forming an image on a recording medium;

an interface means for connecting the image forming apparatus to the external host apparatus; and

a file system emulation means for:

(a) emulating a file system contained in the disk storage apparatus of the external host apparatus, and

(b) making the image forming apparatus look like the disk storage apparatus to the external host apparatus connected to the image forming apparatus to transfer said image from said external host apparatus to said image forming means via said interface means.

4. The image forming apparatus according to claim **3**, wherein said interface means is a small computer system interface (SCSI).

5. The image forming apparatus according to claim **3**, wherein said file system emulation means comprises a memory means for storing data of a predetermined region of the file system, to which region basic information of the file system has been written.

6. The image forming apparatus according to claim **5**, wherein said memory means is a nonvolatile memory.

7. The image forming apparatus according to claim **5**, wherein said memory means is a magnetic storage.

8. The image forming apparatus according to claim **3**, wherein said file system emulation means comprises a parameter setting means for creating a parameter file required to set an operation mode of the image forming apparatus and for setting the operation mode of the image forming apparatus based on parameter data written to the parameter file from the external host apparatus.

9. The image forming apparatus according to claim **8**, wherein said file system emulation means creates an image data file of a size determined by the parameter data after the parameter data have been written to the parameter file by the external host apparatus.

10. The image forming apparatus according to claim **9**, wherein the external host apparatus writes image data to the image data file so that the image data are transferred from the external host apparatus to the image forming apparatus.

* * * * *

Exhibit C

2182 pays 4-8-02

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

RECEIVED
APR 0 5 2002
Technology Center 2100

Patent Application Serial No. 09/331,002

Filing Date: June 14, 1999

For: Flexible Interface

Inventor: Michael Tasler

OIPE JC93 APR 0 3 2002 PATENT & TRADEMARK OFFICE

Group Art Unit: 2182

Examiner: Du, Thuan N.

Docket No.: 13189.129
(Formerly 2055/101)

Paper No.: 8

Certificate of Mailing Under 37 CFR 1.8

I hereby certify that this correspondence, along with all papers referred to as being enclosed or attached, are being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to Box Non-Fee Amendment, Assistant Commissioner for Patents, Washington, DC 20231.

March 18, 2002
Date

Elaine C. VonSpreckelsen
Elaine C. VonSpreckelsen

BOX NON-FEE AMEMDMENT
ASSISTANT COMMISSIONER FOR PATENTS
WASHINGTON, DC 20231

Sir:

This Amendment and Remarks is responsive to the Office Action mailed December 18, 2001.

AMENDMENT

In The Claims:

Please cancel claim 5.

Please amend claims 1, 12, 13 and 15 as follows:

1. (Amended) An interface device for communication between a host device, which comprises drivers for input/output devices customary in a host device and a multi-purpose interface, and a data transmit/receive device, the data transmit/receive device being arranged for providing analog data, comprising:

a processor;

a memory;

a first connecting device for interfacing the host device with the interface device via the multi-purpose interface of the host device; and

a second connecting device for interfacing the interface device with the data transmit/receive device, the second connecting device including a sampling circuit for sampling the analog data provided by the data transmit/receive device and an analog-to-digital converter for converting data sampled by the sampling circuit into digital data,

wherein the interface device is configured by the processor and the memory to include a first command interpreter and a second command interpreter,

wherein the first command interpreter is configured in such a way that the command interpreter, when receiving an inquiry from the host device as to a type of a device attached to the multi-purpose interface of the host device, sends a signal, regardless of the type of the data transmit/receive device attached to the second connecting device of the interface device, to the host device which signals to the host device that it is an input/output device customary in a host device, whereupon the host device communicates with the interface device by means of the driver for the input/output device customary in a host device, and

wherein the second command interpreter is configured to interpret a data request command from the host device to the type of input/output device signaled by the first command interpreter as a data transfer command for initiating a transfer of the digital data to the host device.

11 ~~12~~. (Amended) An interface device for communication between a host device, which comprises a multi-purpose interface and a specific driver for this interface, and a data transmit/receive device, the data transmit/receive device being arranged for providing analog data, comprising:

a processor;

a memory;

a first connecting device for interfacing the host device with the interface device via the multi-purpose interface of the host device; and

a second connecting device for interfacing the interface device with the data transmit/receive device, the second connecting device including a sampling circuit for sampling the analog data provided by the data transmit/receive device and an analog-to-digital converter for converting data sampled by the sampling circuit into digital data,

where the interface device is configured using the processor and the memory to include a first command interpreter and a second command interpreter,

wherein the first command interpreter is configured in such a way that the interface device, when receiving an inquiry from the host device as to the type of a device attached at the multi-purpose interface of the host device, sends a signal, regardless of the type of the data transmit/receive device attached to the second connecting device of the interface device, to the host device which signals to the host device that it is an input/output device customary in a host device, whereupon the host device communicates with the interface device by means of the specific driver for the multi-purpose interface, and

wherein the second command interpreter is configured to interpret a data request command from the host device to the type of input/output device signaled by the first command interpreter as a data transfer command for initiating a transfer of the digital data to the host device.

12 ~~13~~. (Amended) An interface device according to claim 11 ~~12~~, wherein in addition to the first connecting device of the interface device, there is a further input/output device at the multi-purpose interface of the host device, and wherein the interface device can communicate directly with said further input/output device via the specific driver for the multi-purpose interface.

14 ~~15~~. (Amended) A method of communication between a host device, which comprises drivers for input/output devices customary in a host device and a multi-purpose interface, and a data transmit/receive device, the data transmit/receive device being

arranged for providing analog data, via an interface device, comprising:

interfacing of the host device with a first connecting device of the interface device via the multi-purpose interface of the host device;

interfacing of the data transmit/receive device with a second connecting device of the interface device, the second connecting device including a sampling circuit for sampling the analog data provided by the data/transmit/receive device and an analog-to-digital converter for converting data sampled by the sampling circuit into digital data;

inquiring by the host device at the interface device as to the type of device to which the multi-purpose interface of the host device is attached;

regardless of the type of the data transmit/receive data attached to the second connecting device of the interface device, responding to the inquiry from the host device by the interface device in such a way that it is an input/output device customary in a host device, whereupon the host device communicates with the interface device by means of the usual driver for the input/output device, and

interpreting a data request command from the host device to the type of input/output device customary in the host device as a data transfer command for initiating a transfer of the digital data to the host device.

REMARKS

The draftsperson has objected to the drawings. Clean formal drawings responsive to the objections are enclosed.

The Office Action rejected claim 13 under 35 U.S.C. 112, second paragraph, on the basis of insufficient antecedent basis for "the hard disk" in line 4. This has been corrected by amendment.

The Office Action rejected claims 1 – 16 under 35 U.S.C. 103(a) as being unpatentable over Applicant's admission of prior art in view of United States Patent No. 5,499,378 issued to McNeill, Jr. et al. Claims 1, 12, 13 and 15 gave been amended, and this rejection is respectfully traversed with respect to the claims as amended for the

reasons given below. In particular, the data transmit/receive device is arranged for providing analog data. Support for this amendment is in the penultimate paragraph of page 4. Since the second connecting device includes sample and hold circuits 1515 and an A/D converter 1503, it is clear that the data transmit/receive device to be connected to the second connecting device of the subject interface provides analog data. As to the amendment raised with respect to the second connecting device, please refer to FIG. 2 and to the just outlined elements. Regarding the amendment "first command interpreter and second command interpreter", please refer to the last paragraph of page 9. The first command interpreter performs the response to the inquiry from the host device as outlined in the penultimate paragraph of amended claim 1, while the second command interpreter interprets a data request command from the host device; for example, "read file xy" as a data transfer command for initiating a transfer of the digital data to the first device, i.e., of the digital data that have been derived from the analog data of the data transmit/receive device. Support is in the first paragraph of page 10 of the specification (final version to be filed as first preliminary amendment).

McNeill, Jr. et al. discloses a small computer system emulator for non-local SCSI devices. When FIG. 2 is considered, one may compare the initiator with the host device in claim 1, one may compare the magnetic disk 16 to the data transmit/receive device, and one may compare the target device to the interface device in claim 1.

The purpose of McNeill, Jr. et al. is to provide an access to a non-SCSI device via a SCSI bus. In particular, the initiator sends a request to the target, the request being in accordance with the SCSI protocol. The target translates this request into a request suitable for magnetic disk 16 such that the initiator can access to magnetic disk 16 via SCSI commands. This reference does not disclose that the data transmit/receive device is arranged for providing analog data. It is well known that digital data are stored on magnetic disks.

Additionally, the target does not include any sample and hold circuit or any analog-

B

to-digital converter for converting data from magnetic disk 16 into digital data.

In addition, this reference does not include a first command interpreter that, when asked by the host device as to the type of device connected to the interface, lies to the host computer as to the real nature of the data transmit/receive device. In McNeill, Jr. et al., the initiator asks for a hard disk and the target states that there is a hard disk. In other words, the target does not lie as to the true type of the data transmit/receive device. The conversion done in McNeill, Jr. et al. is to transfer SCSI protocol commands into commands understandable for the magnetic disk.

Finally, McNeill, Jr. et al. does not include a second command interpreter as defined in the last paragraph of amended claim 1.

For the above reasons, McNeill, Jr. et al. is not pertinent to the present invention as defined in amended claim 1.

We also note that the examiner did not reject claim 5 in detail. Claim 5 includes the A/D converter feature that has been included into amended claim 1. Thus, this also supports the fact that the amended claims are patentable.

For the above reasons, the claims as amended are believed to be patentable and their reconsideration and allowance are respectfully requested. It is believed no fee is due. If any fees are due, the Commissioner is authorized to charge them to Deposit Account No. 50-1848.

Respectfully submitted,
PATTON BOGGS LLP

By: _____

Leslie S. Garmaise, Reg. No. 47,587
Telephone: (303) 379-1131
Facsimile: (303) 379-1155
Customer No.: 24283

Version With Markings To Show Changes Made

In The Claims:

Please cancel claim 5.

Please amend claims 1, 12, 13 and 15 as follows:

1. (Amended) An interface device for communication between a host device, which comprises drivers for input/output devices customary in a host device and a multi-purpose interface, and a data transmit/receive device, the data transmit/receive device being arranged for providing analog data, comprising [the following features]:

a processor;

a memory;

a first connecting device for interfacing the host device with the interface device via the multi-purpose interface of the host device; and

a second connecting device for interfacing the interface device with the data transmit/receive device, the second connecting device including a sampling circuit for sampling the analog data provided by the data transmit/receive device and an analog-to-digital converter for converting data sampled by the sampling circuit into digital data,

wherein the interface device is configured by the processor and the memory to include a first command interpreter and a second command interpreter,

wherein the first command interpreter is configured in such a way that the command interpreter, when receiving an inquiry from the host device as to [the] a type of a device attached to the multi-purpose interface of the host device, sends a signal, regardless of the type of the data transmit/receive device attached to the second connecting device of the interface device, to the host device which signals to the host device that it is an input/output device customary in a host device, whereupon the host device communicates with the interface device by means of the driver for the input/output device customary in a host device, and

wherein the second command interpreter is configured to interpret a data request command from the host device to the type of input/output device signaled by the first

B

command interpreter as a data transfer command for initiating a transfer of the digital data to the host device.

12. (Amended) An interface device for communication between a host device, which comprises a multi-purpose interface and a specific driver for this interface, and a data transmit/receive device, the data transmit/receive device being arranged for providing analog data, comprising [the following features]:

a processor;

a memory;

a first connecting device for interfacing the host device with the interface device via the multi-purpose interface of the host device; and

a second connecting device for interfacing the interface device with the data transmit/receive device, the second connecting device including a sampling circuit for sampling the analog data provided by the data transmit/receive device and an analog-to-digital converter for converting data sampled by the sampling circuit into digital data,

where the interface device is configured using the processor and the memory to include a first command interpreter and a second command interpreter,

wherein the first command interpreter is configured in such a way that the interface device, when receiving an inquiry from the host device as to the type of a device attached at the multi-purpose interface of the host device, sends a signal, regardless of the type of the data transmit/receive device attached to the second connecting device of the interface device, to the host device which signals to the host device that it is an input/output device customary in a host device, whereupon the host device communicates with the interface device by means of the specific driver for the multi-purpose interface, and

wherein the second command interpreter is configured to interpret a data request command from the host device to the type of input/output device signaled by the first command interpreter as a data transfer command for initiating a transfer of the digital data to the host device.

13. (Amended) An interface device according to claim 12, wherein in addition

JB

to the first connecting device of the interface device, there is a further input/output device at the multi-purpose interface of the host device, and wherein the interface device can communicate directly with [the hard disk] said further input/output device via the specific driver for the multi-purpose interface.

15. (Amended) A method of communication between a host device, which comprises drivers for input/output devices customary in a host device and a multi-purpose interface, and a data transmit/receive device, the data transmit/receive device being arranged for providing analog data, via an interface device, comprising [the following steps]:

interfacing of the host device with a first connecting device of the interface device via the multi-purpose interface of the host device;

interfacing of the data transmit/receive device with a second connecting device of the interface device, the second connecting device including a sampling circuit for sampling the analog data provided by the data/transmit/receive device and an analog-to-digital converter for converting data sampled by the sampling circuit into digital data;

inquiring by the host device at the interface device as to the type of device to which the multi-purpose interface of the host device is attached;

regardless of the type of the data transmit/receive data attached to the second connecting device of the interface device, responding to the inquiry from the host device by the interface device in such a way that it is an input/output device customary in a host device, whereupon the host device communicates with the interface device by means of the usual driver for the input/output device, and

interpreting a data request command from the host device to the type of input/output device customary in the host device as a data transfer command for initiating a transfer of the digital data to the host device.

Exhibit D

By the Editors of the
AMERICAN HERITAGE® Dictionaries

AN A TO Z GUIDE
TO HARDWARE,
SOFTWARE,
AND CYBERSPACE

dictionary of computer *and* internet *words*

Words are included in this Dictionary on the basis of their usage. No investigation has been made of common-law trademark rights in any word, because such investigation is impracticable. The inclusion of any word in this Dictionary is not, however, an expression of the Publisher's opinion as to whether or not it is subject to any proprietary rights. Indeed, no definition in this Dictionary is to be regarded as affecting the validity of any trademark.

Copyright © 2001 by Houghton Mifflin Company.
All rights reserved.

No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or by any information storage or retrieval system without the prior written permission of Houghton Mifflin Company unless such copying is expressly permitted by federal copyright law. Address inquiries to Reference Permissions, 222 Berkeley Street, Boston, Massachusetts 02116.

Visit our website: www.houghtonmifflinbooks.com

Library of Congress Cataloging-in-Publication Data

Dictionary of computer and internet words : an A to Z guide to hardware, software, and cyberspace.
p. cm.
ISBN 0-618-10137-3
1. Computers--Dictionaries. 2. Internet--Dictionaries.
QA76.15 .D5255 2001
004'.03--dc21

2001016890

Manufactured in the United States of America

DOH 10 9 8 7 6 5 4 3 2 1

from one system to another as a unit. When a block is created, additional information is usually included that can be used to check if the block was garbled during transmission. In general, the larger the block size, the faster the data transfer rate, but if noisy telephone lines or other disruptions cause errors that make it necessary to repeat blocks, smaller block sizes are more efficient. The most common transfer protocols use block sizes from 128 bytes to 1,024 bytes.

block graphics Onscreen graphic images that are made of characters in the ASCII extended character set having the decimal values 176 through 223. These characters consist of rectangles and horizontal and vertical lines of various thicknesses, sizes, and shadings. Being elements in the character set, they are processed in the same way that other characters, such as the letters and numbers, are. As a result, block graphics are processed and displayed more rapidly than bit-mapped graphics. However, images created with block graphics are more limited and are low-resolution. *See the IBM Extended ASCII Character Set table at* **ASCII**.

block protection In word processing, a feature that allows a block of text to be kept together as a unit when moved to a location where it would otherwise be interrupted by page breaks.

.BMP A file extension in MS-DOS for files with bit-mapped graphics.

BNC connector A connector used to connect coaxial cables. By placing a cable having this connector into the end of another, you can lock them together by turning a ring that surrounds the end of the connector. *See illustration.*



BNC CONNECTOR A male BNC connector

board *See* **printed circuit board**.

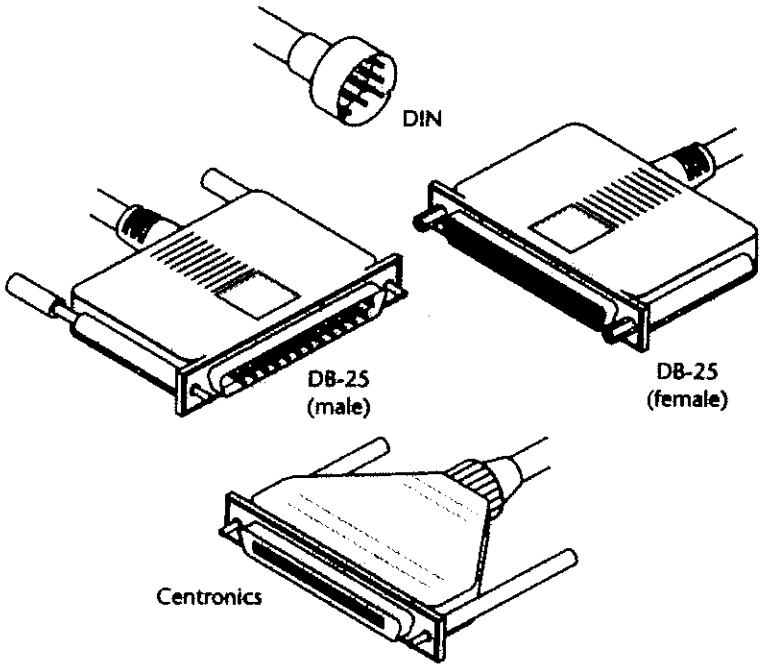
boilerplate Text or graphical elements that need to be used frequently in numerous documents, such as standard language required in mortgage documents, or a company logo and letterhead. By saving boilerplate to a memory device, you can copy it into any documents or programs that require its use.

boldface A typeface in which the letters are heavier and darker than normal. The entry words in this dictionary are printed in boldface. *See illustration at* **font family**.

CONFIG.SYS. Whenever you ask your computer to run a program, it checks the configuration file to find the current parameters for that program. In a word processing program, for example, the parameters would include the current settings for line spacing, margin width, and font specifications.

connectivity The ability to make and maintain a connection between two or more points in a telecommunications system.

connector A coupler used to join two cables or to plug a cable into a port or interface. Types of connectors include the DIN connector and the DB connector. *See illustration. See also illustration at* **printed circuit board.**



CONNECTOR

connect time The elapsed time during which a user is connected to a remote terminal, especially that of an Internet service provider.

console The portion of a computer that allows you to communicate with the CPU (central processing unit). In a personal computer, the console is made up of the keyboard, the mouse, and the monitor. In a network, the console is the terminal that controls the mainframe or server.

constant A value, such as a number or a string, that does not change. Programs generally use both constants and variables.

Exhibit E

By the Editors of the

American Heritage® Dictionaries

Computer Words

An A to Z Guide to Today's Computers

Words included in this Dictionary that are known to have current trademark registrations are shown with initial capital and are also identified as trademarks. No investigation has been made of common-law trademark rights in any word, because such investigation is impracticable. The inclusion of any word in this Dictionary is not, however, an expression of the Publisher's opinion as to whether or not it is subject to proprietary rights. Indeed, no definition in this Dictionary is to be regarded as affecting the validity of any trademark.

Copyright © 1995 by Houghton Mifflin Company. All rights reserved.

No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or by any information storage or retrieval system without the prior written permission of Houghton Mifflin Company unless such copying is expressly permitted by federal copyright law. Address inquiries to Reference Permissions, 222 Berkeley Street, Boston, Massachusetts 02116.

For information about this and other Houghton Mifflin trade and reference books and multimedia products, visit The Bookstore at Houghton Mifflin on the World Wide Web at (http://www.hmco.com/trade/).

Library of Congress Cataloging-in-Publication Data

Dictionary of computer words.—Rev. ed.
p. cm.
Includes index.
ISBN 0-395-72834-7 (acid-free paper)
1. Computers—Dictionaries.
QA76.15.D5259 1995
004'.03—dc20 95-1175
CIP

Manufactured in the United States of America

DOH 10 9 8 7 6 5 4 3 2 1

Book design by Anne Chalmers

ART CREDITS: *Apple Computer:* desktop, dialog box, keyboard (Apple Adjustable), menu, overlaid windows, toolbar; *Fountain Hills Systems Inc.:* keyboard (Ergonomic Keyboard); *Lexmark International, Inc.:* keyboard (Select-Ease); *Library of Congress:* pixel (photograph); *Lotus Development Corporation:* spreadsheet; *Maureen Wilken/Cheryl Snyder:* range; *Microsoft Corporation:* screen shots at the entries alert box, character-based, graphical user interface, and range reprinted with permission from Microsoft Corporation; illustration of the Natural Keyboard at the entry keyboard reproduced with permission from Microsoft Corporation; *Tech-Graphics:* antialiasing, Bézier curve, chip, computer, connector, DIP switch, Dvorak keyboard, floppy disk, hard disk, hierarchical, landscape, letter-quality, mouse, network, outline font, overlaid windows, pixel, printed circuit board, QWERTY keyboard, sector, sine wave, software, trackball, write-protect; *U.S. Environmental Protection Agency:* Energy Star.

Acl

Pre

Dic

Sul

each other. For example, the operating system needs to know whether you're going to be using a mouse and what type of *printer* the computer is hooked up to.

On DOS-based computers, the system is configured by placing commands in the file *CONFIG.SYS*. In OS/2 systems, many settings are saved in binary *INI files*. On Macintosh computers, the system's parameters are set with the *Control Panel*, a Macintosh *desk accessory*.

configuration file A file that tells a computer what *configuration* to use for a specified *program*. The configuration file for *DOS* is called *CONFIG.SYS*. Whenever you ask your computer to run a program, it checks the configuration file to find the current *parameters* for that program. In a word processing program, for example, the parameters would include the current settings for line spacing, margin width, and *font* specifications.

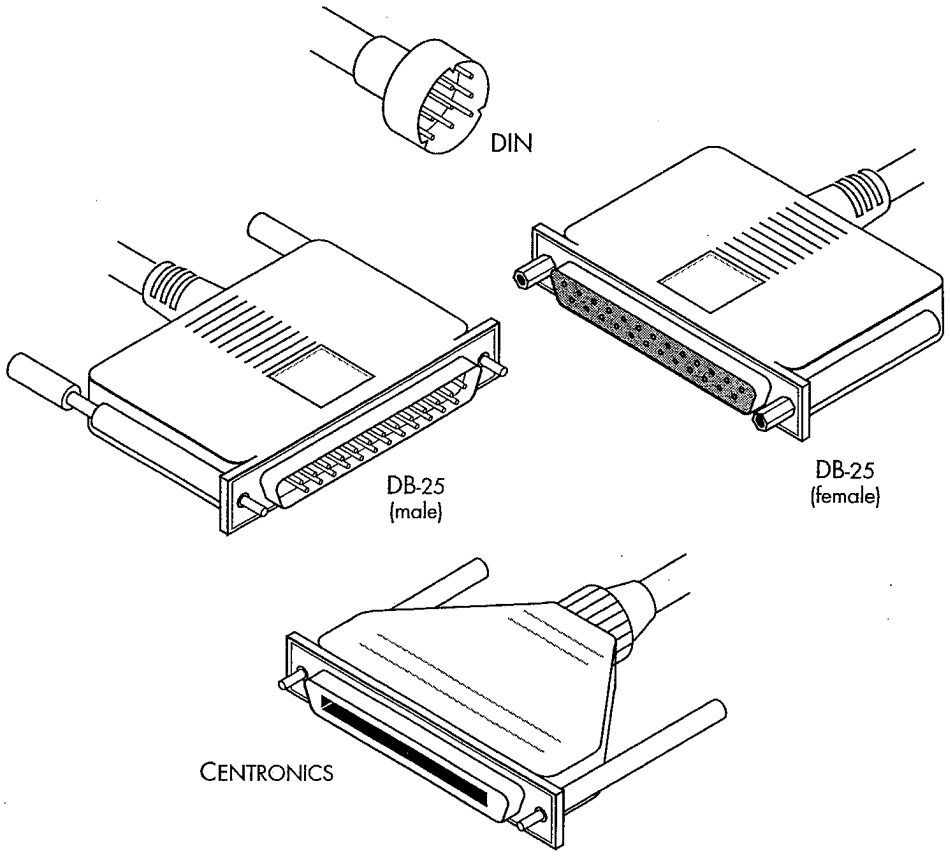
connector A coupler used to join two cables or to plug a cable into a *port* or *interface*. There are several types of connectors. Three common ones are the *DIN connector*, the *DB connector*, and the *Centronics interface*. See illustration. See also illustration at *printed circuit board*.

console The portion of a computer that allows you to communicate with the *CPU* (central processing unit). In a personal computer, the console is made up of the *keyboard* and the *monitor*. In a network, the console is the *terminal* that controls the *mainframe* or *server*.

constant A value, such as a number or a *string*, that does not change. *Programs* generally use both constants and *variables*.

context-sensitive help A mode of *help* in which the primary information available to the user varies according to the operation the user is performing.

context switching See **task switching.**



connector
Several kinds of connector

continuous tone A method of creating an image out of smoothly gradated colors or shades of gray, as in photography. Color *printers* and some color *graphics* programs use continuous tone, but some black-and-white printers and most black-and-white graphics software use patterns of black dots of variable size and density to simulate continuous tone. See also *halftone*.

control character A keyboard *character* that serves, usually in combination with the *Control key*, to give the computer a *command* when both keys are pressed together. For example, in some *word processing* programs the control character *x* means "delete" and the control character *c* means "copy."

Exhibit F

Case 1:07-mc-0 This material may be protected by Copyright law (Title 17 U.S. Code) Page 2 of 27

William E. Ham

Recent Advances in Basic Physical Technology for Parallel SCSI: UltraSCSI, Expanders, Interconnect, and Hot Plugging

DIGITAL uses SCSI technology in most of its storage products and consequently has led major standards and industry bodies to improve the technology in the following areas: increased synchronous data phase speed beyond fast SCSI; longer, more complex electrical configurations by means of expander circuits; versatile and more manageable connectivity through a smaller, improved physical interconnect; and dynamic device insertion and removal. Data phase transmission rate extension is achieved through understanding and controlling silicon chip timing and transmission media parameters. Using expander devices to confine transmission line effects to shorter segments allows large increases in the maximum distance between devices and in the device population within the same SCSI domain. Expanders enable complex, hublike configurations to be created without changing existing SCSI devices or software. The use of 0.8-millimeter connector technology and consideration of cable losses has reduced the physical size of the external shielded interconnect by approximately two thirds, decreased the number of parts required to support complex configurations by a factor of 10, and increased the interconnect density to the same level used in serial SCSI. Finally, the mating and demating events that occur during device insertion and removal produce a spectrum of small, undetectable, electrical disturbances on the active bus that appear to be limited by the physics of the media and device capacitance.

Introduction

Parallel Small Computer System Interface (SCSI) is the workhorse technology for most of the storage applications in DIGITAL products today. This device and interconnect technology spans all system offerings from the simplest to the most complex. SCSI was introduced to the higher-end products in the early 1990s as the open systems follow-on to the DIGITAL proprietary Digital Storage System Interconnect (DSSI) and Computer Interconnect (CI) technologies.

As system demands have increased, SCSI has evolved to meet the needs. DIGITAL has made considerable contributions to the technology and led the effort to achieve industry standardization. This paper details the most significant developments in the physical features of parallel SCSI technology over the last several years that have allowed it to continue to serve DIGITAL customers in an effective, competitive way. The discussion targets the following four important areas:

1. Speed increases in the synchronous data phase, which resulted in the ANSI definition of UltraSCSI (Fast-20 SCSI) technology[1]
2. Development of software-invisible circuits, generally called expanders, that enable segmentation of SCSI domains into easily managed pieces
3. New connector and cable technology, namely the Very High Density Cabled Interconnect (VHDCI) device, that decreases the interconnect size and complexity by many fold[2]
4. Dynamic removal and replacement of devices on an active bus, which is referred to as hot plugging

DIGITAL made substantial contributions in the four areas. This work included creating the expander and interconnect standards projects; leading the working groups that defined the Fast-20, expander, and interconnect standards; providing data for the Fast-20 and hot-plugging projects; and proposing and gaining approval for the hot-plugging standard.

The author has taken a phenomenological approach throughout, because in most cases there are too many unknowns to achieve a rigorous analytical result. This

paper focuses on developments from SCSI-2 through UltraSCSI and specifically does not address the new Low Voltage Differential (LVD) technology being introduced for the highest-speed applications.

Pedigree

SCSI is defined in several ANSI standards[1,3,4] and in the material that was developed to create these standards.[5,6] The standards were generated over the last decade through a cooperative effort of approximately 60 major companies in the computer and computer support industry. As a result of this pedigree, the prime directive for SCSI technology is interoperability of devices designed and manufactured by different companies.

The details of the physical designs used to implement SCSI may not be visible to users and researchers; these details contain much of the marketing and technical differentiation between the products of the participating companies and are therefore hidden in the silicon design. The behavior at the device connector pervades the SCSI specifications. The basic assumption is that as long as the properties are compatible at these connectors, device substitution is possible. Thus, SCSI devices may be both interoperable and of different designs.

Basic Architecture

This section reviews the basic architecture of parallel SCSI. The SCSI bus is a parallel, multidrop, wired-OR configuration.

Signal Multiplexing and Phases The parallel signal construction of the bus allows multiplexing of some signals during different phases of communication so that the same signal lines may have very different functions in different phases. The physical behavior of signals is usually limited by the phase during which the shortest pulses are used and the demands for signal integrity are the highest. The limiting SCSI phase is the data phase (payload phase) that is executed with the highest synchronous rate. For UltraSCSI, this peak repetition rate is 20 megahertz (MHz). Table 1 contains the generally accepted terminology related to data phase speeds.

Because of the wired-OR property, each signal in the bus must be driven to a known state even if no SCSI device is actually driving the signal. SCSI uses the logical 0 state (negated state) as the undriven state and uses the bus terminators to drive the signal to this state in the absence of any driving devices. The device signal drivers must overcome this terminator-driven logic state of 0 in order to send a logical 1 (asserted state) onto the signal line.

SCSI signals must support all frequencies, from statically driven by the terminators only (DC) to the third harmonic of the fastest signal edge in the synchronous data phase. In many cases, the same wire must support all these frequencies at different times during the SCSI protocol.

The highest signal edge slew rates for UltraSCSI are approximately 500 millivolts per nanosecond (mV/ns). A 2-volt (V) transition requires approximately 4 ns/5.4 ns/meter (m) = 0.74 m for a signal edge (assuming 5.4 ns/m as the propagation velocity of the signal edge). Therefore, some relief exists because the connectors and cable assembly terminations are much smaller than the signal edge length; the connectors and terminations do not need to have carefully controlled characteristic impedance properties. This allows the use of the technology available in the connector and cable assembly industry to optimize the interconnect properties without the considerable design, manufacturing, and test burden imposed by controlled impedance requirements.

Transmission Modes The transmission mode of a SCSI bus is determined by the properties of the terminators that, by definition, constitute the ends of the bus. Terminators also supply most of the energy required to operate the single-ended transmission-mode devices and additionally provide the required matching

Table 1
Terminology for Data Phase Speeds

Data Phase Speed Name	Maximum Transfer Rate (Million transfers/second)[1]	Maximum Byte Rate (Narrow) (Megabytes/second)	Maximum Byte Rate (Wide) (Megabytes/second)
Asynchronous	Unspecified	Typically ~ 3	Typically ~ 6
Slow (synchronous)	5	5	10
Fast (synchronous)	10	10	20
Ultra (synchronous)[2]	20	20	40
Ultra2 (synchronous)[3]	40	40	80
Ultra3 (synchronous)[4]	80 to 100	80 to 100	160 to 200

[1]One transfer is 1 byte in narrow mode and 2 bytes in wide mode; 1 byte equals 8 data bits plus 1 parity bit.
[2]Ultra is synonymous with Ultra1 and Fast-20.
[3]Ultra2 is synonymous with Fast-40.
[4]Rates not yet finalized; Ultra3 is synonymous with Fast-80 or Fast-100.

to the characteristic impedance of the transmission line. In differential SCSI, the terminators provide a small portion of the overall energy required to operate the bus; the differential drivers supply the remainder of the energy.

Drivers that want to transmit an asserted state must overcome the biasing provided by the terminators. The drivers operate locally on the bus and alter the state in their immediate vicinity when they switch on and off. For single-ended SCSI, the 0 state is approximately 2.5 V and the 1 state is approximately 0.5 V. For high-voltage differential SCSI, the 0 state is approximately –1 V to –2 V, and the 1 state is approximately 2 V. (The difference between a state 1 and a state 0 is higher with differential—typically, approximately 4 V.)

For single-ended transmissions, the drivers operate on energy previously stored in the bus by the terminators. This energy is mostly electrostatic energy in the charge stored in the capacitance of the transmission line for negated states and electromagnetic energy in the current flowing through the inductance of the transmission line for asserted states. Ultimately, the terminators will set the state back to negated after the drivers cease to source or sink current; however, this only happens after the round-trip propagation delay from the driver to the farthest terminator if the bus does not have matched characteristic impedance properties.

Approximately the same energy transformations occur for differential SCSI, but significant current is supplied by the drivers for both the asserted and the negated states.

Multidrop Requirements The multidrop architecture requires a continuous low-resistance path called the bus path between the terminators and allows devices to be attached to this path. The number and properties of these attached devices vary widely because of many factors including the speed of operation, the overall length of the bus, and the transmission mode. Attached devices always disturb the transmission line properties of the bus path; the key to successful operation is in the management of the magnitude of these disturbances.

Generally, the more capacitance or electrical length the device has, the more disruptive it is. Placing devices too close together along the bus path can cause them to appear electrically as a single super disruptive device. Placing them too far apart can result in an overall bus length that is too long.

Wired-OR Glitches During the arbitration phase, when the SCSI devices decide which devices will be sending payload data to or from each other, multiple devices may assert the same control line (BSY) at the same time. Each device that wishes to communicate asserts both the BSY line and its respective device identification (ID) line. After examining the asserted ID lines to determine which device has the highest ID, all but the device with the highest ID release the BSY line. This leaves only one device, the winner, asserting the BSY line. While the current in the BSY line is readjusting itself from a multiple-driver asserted condition to a single-driver asserted condition, noise pulses (called wired-OR glitches) propagate throughout the length of the signal line and may be detected collectively as an erroneous phase. Therefore, one of the architectural limits for parallel SCSI is the time required for these wired-OR glitches to settle. This bus settle time is set by protocol at 400 ns and must be interpreted as a round-trip propagation time when using a simple SCSI bus. Allowing some time for propagation through driver and receiver chips yields a maximum physical length for a simple bus of 25 meters.

Areas of Improvement

Thus, the opportunities for improving SCSI derive from appropriately managing the transmission lines, taking advantage of the multidrop architecture offered by a parallel wired-OR structure, using state-of-the-art technology from the interconnect and silicon industry, and making innovative use of the time required for the wired-OR glitches to settle. These techniques are the basis of the development by DIGITAL in the four areas addressed in this paper.

Speed increases in the synchronous data phase are based primarily on increasing the timing precision in the silicon transceivers by using newer silicon technology. The interconnect properties remain largely unchanged from those used for fast SCSI.

Circuits that enable segmentation of SCSI domains into easily managed pieces are based on systematic isolation of transmission line properties and use of wired-OR noise pulse properties. No software, interconnect, or device changes needed to use these circuits.

New connector and cable technology is based on an innovative 0.8-millimeter (mm) ribbon-style connector technology that optimizes the total SCSI electrical requirements with the capabilities of cable and connector design.

Dynamic removal and replacement of devices on an active bus, i.e., hot plugging, is based on the multidrop architecture, which enables devices to be added or replaced without affecting continuity between other devices. Hot plugging depends on understanding and managing the electrical disturbances created during the insertion or removal.

The remainder of this paper provides details of these four areas of improvement. The end result of these extensions to the basic physical architecture of parallel SCSI is a major increase in its capabilities, accompanied by only a very minor disturbance to the installed base, especially the software.

Increasing the Synchronous Data Phase Speed

Beginning with the SCSI-2 standard, the synchronous transmission mode is available for transferring payload data between SCSI devices. The devices select this mode by mutual agreement before any synchronous data is passed. The agreement is achieved by using the asynchronous transmission mode, which is slow but usually reliable.

The synchronous data phase uses the DATA and PARITY bit lines for the data and either the REQ or the ACK control line as a signal that the receiver uses for capturing the data. The term synchronous derives from a specified timing relationship between the bit line signal edges and the REQ or ACK signal edges. (The falling edge of the ACK signal is used when the data phase transmission originates from the SCSI initiator, and the falling edge of the REQ signal is used when the transmission originates from the target.) There is no synchronous relationship between the internal timing references on different SCSI devices, so the receiver must buffer the received data before introducing the data into its internal data management structure. This buffering is usually accomplished by means of a first in first out (FIFO) circuit that uses the REQ or the ACK signal as the latching signal for the incoming data. For convenience, in this paper we only refer to the ACK signal, with the understanding that the same discussion applies to the REQ signal when it is used as the data-latching signal.

Since only the falling edge of the ACK signal is used in the presently specified SCSI versions and an ACK signal is required for every data transfer, it follows that the ACK signal cycles at least twice as fast as the data bits. When a continuous stream of transfers is transmitted, the ACK signal is a regularly repeating signal, nominally, a square wave. An alternating 1/0 pattern produces the highest fundamental frequency for the data bits at half the frequency of the ACK signal. Therefore, the ACK signal requires careful attention since it is the most demanding on the transmission process.

The focus of this section is to examine how the speed of the synchronous data phase was increased by a factor of two to achieve the Fast-20 (UltraSCSI) specification.

Status before UltraSCSI

In 1993, the SCSI-2 standard[3] had been in place for two years, and a follow-on standard called SCSI-3 Parallel Interface (SPI)[4] was technically stable. SPI had been created largely because the specifications in the SCSI-2 standard were not effective in implementing the single-ended version of the synchronous transmission (10 megatransfers per second). The differential version specified in SCSI-2 worked well but was much more expensive in cost, power, and space than the single-ended version. Therefore, most of the interest was in making the fast single-ended version work adequately.

Taking single-ended SCSI from asynchronous and slow synchronous (5 megatransfers per second) to the fast synchronous technology was difficult. The prevailing opinion was that the SPI standard represented the final improvement to parallel SCSI. This view set the stage for a number of alternate physical technologies based on the serial point-to-point transmission schemes used in communications technologies, e.g., Fiber Distributed Data Interface (FDDI) and Ethernet, to be used for higher-performance storage applications.

DIGITAL's Storage Bus Technical Office had seen many instances of difficult implementations that were the result of less-than-optimal understanding and management of the specification margins. No credible study had been presented on the margins available in SCSI, so the thrust was to create baseline characteristics of multidrop parallel SCSI to determine where unused margin might exist.

Little data was available on the precise reasons why specific implementations of fast synchronous SCSI did not work. The system would hang or report various error messages with almost no indication of the basic causes. A method that could report margin to failure and mechanism of failure was needed to unravel this situation. Therefore, the approach DIGITAL took was to step back from full SCSI implementations and to examine the pieces without the encumbrance of the SCSI protocol.

One of the most mysterious areas was the behavior of SCSI receivers. The SCSI-2 and SPI specifications used bipolar transistor-transistor logic (TTL) levels as the basic receiver input levels. Almost all SCSI devices were being designed with complementary metal-oxide semiconductor (CMOS) technology, so the differences between the receiver properties presented a key opportunity for hidden margin. Other unknown areas were jitter, cross talk, skew, ground offset, effects of stubs, and worst-case configurations.

DIGITAL built a special test environment to systematically examine each piece of parallel SCSI. The environment was named the PBDIT, an acronym for parallel bus data integrity tester. This test environment made it possible to systematically examine the real margins to failure for the key pieces and to develop the confidence that SCSI could be used at elevated speeds and be made highly robust at the slower speeds.

Special Test Environment

The test environment was built to allow known data patterns to be transmitted across a SCSI device, into SCSI transmission media, and then into another SCSI device. The same data pattern is loaded into both sides so the receiver knows exactly what data it is supposed

to receive. The transmitting side is called the exciter, and the receiving side is called the comparator. Received data is committed to the comparator by using one bit line as the latching ACK signal in a manner exactly like that specified in synchronous SCSI transmissions. The test environment allows the position of the ACK signal to be adjusted with respect to the data signal edges.

Since the comparator knows the data pattern that is transmitted, it is possible to isolate the precise data bit that caused the transmission error. This kind of error-directed methodology has found widespread use in the integrated circuit industry.

Other features of this test environment include detachable load boards that contain the SCSI drivers, terminators, receivers, connectors, or any other physical media-dependent components. The minimum requirements for a load board are that the exciter contain the SCSI driver and a connector and that the comparator contain the SCSI receiver and a connector. Other components may be placed between the load boards for different test conditions. The SCSI driver must have accessible points for the exciter logic, and similarly, the SCSI receiver must have output points to drive the comparator. These requirements eliminate drivers and receivers that are imbedded within chips with other functions. Fortunately, separate SCSI drivers are available for both single-ended and differential versions. (The differential versions normally use separate chips, but only a few choices are presently available for the separate single-ended versions.)

The test environment is useful for developing the understanding of operating mechanisms and for measuring the margins for specific hardware configurations. This environment is not useful for deriving specifications, since the performance at the specified interfaces, i.e., the device connectors, is not directly observable. Oscilloscope measurements provide the basis for setting compliance specifications, since these measurements can be performed at the connectors. The basic question that needed an answer was, Can parallel SCSI be operated at elevated speeds with reasonable margin to failure? DIGITAL optimized the special test environment to answer this question. Other specifications that would be necessary to ensure interoperable operation between UltraSCSI devices could be derived if it appeared possible to achieve the end result.

The data pattern loading and digital control of the exciter and the comparator were achieved through optically coupled means. This allowed the ground offset voltage to be adjusted between the driver and the receiver without compromising the operation of the logic.

The data flows only from the exciter to the comparator. If bidirectional information is desired, the physical connections between the exciter and comparator have to be reversed. This scheme leaves untested the cross-talk effects on the REQ signal that is traveling in the opposite direction to the ACK signal (if ACK is synchronized with the data as in a write operation). Separate measurements are necessary to examine this issue. Cross talk into other control lines is addressed by holding these lines constant in the data pattern transmitted.

The SCSI standard deals with the REQ cross-talk issue by requiring that the data lines be physically separated from the REQ and ACK lines in the transmission media. Measurements not reported in this paper have confirmed negligible speed-related cross talk into the REQ line.

Up to 27 pairs of 3-byte-wide lines (wide SCSI uses only 18 pairs for high-speed transmissions) can be tested with the special test environment. Figure 1 is a functional diagram of the test environment. The SCSI terminators are shown as separate from the load

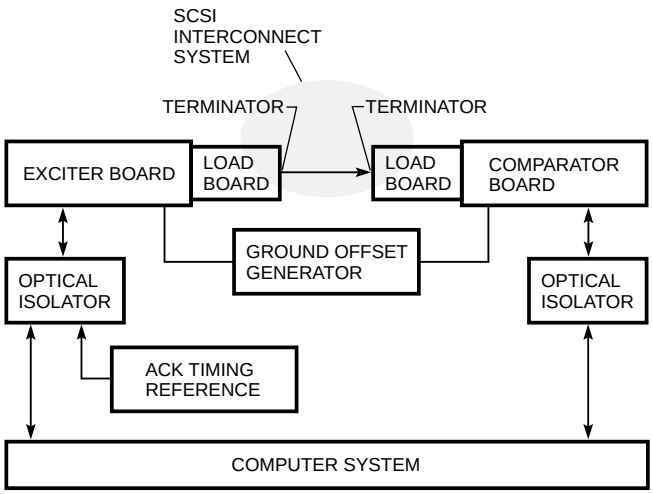


Figure 1
Special Test Environment

boards in this case. A key feature of this kind of testing is that the test does not necessarily stop when an error is detected. In fact, the environment may detect errors 100 percent of the time. This acceptable behavior allows mapping of the complete bit-error response of the system.

Sample Data from the Special Test Environment The test environment allows a multitude of tests to be performed. The test scheme described in this section is the one that was used to establish the basic timing margins available from normal SCSI silicon, cables, connectors, and terminators.

A random repeating data pattern with 16 thousand different bit combinations was used as the basic data pattern. This pattern was transmitted over a period of time, and the number of errors detected was recorded. In this test, an error is defined as one or more bits in the received data transfer that do not match the transmitted bit. To acquire a new error rate data point, the transmission test is repeated by using exactly the same number of transfers in the same time period with the same data pattern but with some test parameter changed.

Virtually any parameter can be varied for different tests. For a given physical configuration, the most useful parameter for determining the timing margin is the position of the ACK pulse with respect to the data edges. The basic data then becomes the number of errors detected and the position of the ACK pulse edge.

There are two basic random variables operating in this scheme: the data pattern and the jitter induced by non-data-dependent sources. It is easy to separate these two variables by using extremes in the data pattern: very few transitions and the maximum number of transitions (every data edge has a transition, i.e., alternating 1/0 pattern). Although this level of precision is available, we will see that we really do not need to bother for parallel SCSI at the maximum UltraSCSI rate.

Figure 2 shows a typical error rate plot from a simple single-ended configuration made from ordinary SCSI interconnect hardware and transceivers being tested at the maximum UltraSCSI rate. Each data point represents a 3-second sample (60 million transfers) at each ACK position. The ACK position is incremented in 0.1-ns steps for a total of 240 independent tests in the plot. To minimize the testing time, we tested only the time ranges from –3 to 9 ns and 44 to 56 ns. The individual data points are not distinguishable in this presentation, and there is very little scatter between neighboring points. In Figure 2, the error rate of 1 is used to indicate that no errors were detected, since the log of 0 is not easy to plot.

Examination of the raw data reveals that the plot is monotonic in detected error rate to the fourth decimal place. This indicates an extremely predictable situation as far as behavior of the same set of hardware is concerned. That is, there is virtually no Gaussian jitter present, and a SCSI system could be designed to be quite reliable and stable at the maximum UltraSCSI rate.

Extending the sample period to 5 minutes made no difference in the position of the key features. Using the 3-second sampling time, the entire data set could be acquired automatically in approximately 12 minutes.

The onset of errors is extremely sharp as the ACK position approaches the critical position. One hundred picoseconds changes the observation from 0 to 864 errors near the 8-ns position. On the other end, the 50.1-ns time produced 7 errors, and the 50.2-ns time produced 425 errors. No errors were detected at any of the times between 50.1 ns and 7.9 ns. This data shows that there are no strange effects that prevent SCSI from operating at the maximum UltraSCSI rate.

As the ACK position proceeds into the region of more errors, a condition is finally reached in which *all* the transfers have errors. On the one hand, the probability that one transfer has the same data content as its

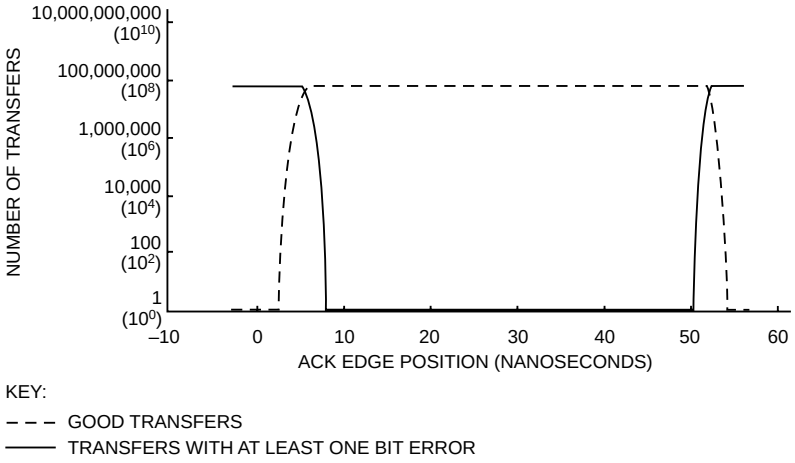


Figure 2
Typical UltraSCSI Error Rate Plot

neighbor's is very small with this random data pattern. On the other hand, since a random data pattern is being used, there is a reasonable chance that a bit will actually match that transmitted in one state but not in the other state. The random data pattern tends to spread out the time between the first error and the last good transfer. In the limit, for perfectly random data, this time is a measure of the total timing imprecision in the system.

This imprecision includes skew in the exciter and comparator boards, in the SCSI drivers and receivers, and in the cable transmission media (including loads, if any), and all forms of jitter. For the test conditions shown in Figure 2, the total difference is 3.6 ns near the 5-ns point and 5.4 ns near the 52-ns point. This shows that the skew specifications in the SCSI standard are overspecified as compared to actual hardware performance.

The data shown in Figure 2 is representative of a large variety of configurations up to approximately 3 meters long and loaded or up to much longer point-to-point lengths (20 meters or more [see Figure 6]). The error-free window can be made to collapse by adding too many loads or by using the wrong impedance cable, improper terminators, receivers with the wrong threshold voltages, or other bus component and configuration parameters. However, the details of the actual hardware and configuration do not affect the basic conclusion derived from Figure 2, namely, that a great deal of timing margin is available at the maximum UltraSCSI rate when ordinary SCSI hardware is used.

To put this into perspective, basic gigabit-per-second serial transmissions with approximately twice the basic bandwidth of UltraSCSI have bit times of about 1 ns and timing margins of a few hundred picoseconds. UltraSCSI has an effective margin window of a few tens of nanoseconds. This represents two orders of magnitude more margin for the parallel SCSI application.

The initial errors usually originate from the same bit. This bit is the one with the most unfavorable timing skew with respect to the ACK signal. The cliff is not perfectly sharp because there is a 50 percent chance that the data transmitted is the same as that expected even under the error case and, more importantly, because there is some level of jitter present. It is this jitter that softens the cliff. Thus, the first errors detected happen when the skew of the weakest bit adds to the tail of the jitter distribution. Only a few errors are present because only a small part of the jitter population extends far enough to trigger the error. SCSI systems will experience virtually no errors because of these mechanisms in service if one operates 1 ns or more away from an error cliff.

Note that these results from the special test environment almost always yield margins higher than those calculated from a set of interoperability specifications. This is because the interoperability specifications must allow margin for each piece, and the special test environment reports the integrated result from many pieces in the complete SCSI connection.

Higher Speeds The main effect of further increasing the transfer rate above the maximum UltraSCSI rate in the same set of hardware is to change the time position of the onset of nonzero error rates and to narrow the error-free region. Figure 3 shows an example of data from Fast-40 transmissions using separate high-voltage differential transceivers on each bit. (This data was acquired by DIGITAL's Storage Bus Technical Office in 1994.)

The error-free zone has narrowed to approximately 15 ns, and the time between first error and 100 percent errors has widened on both sides, but still no uncontrolled regions exist. This strongly suggests that at least Fast-40 transfer is possible with no major technology changes in the interconnect.

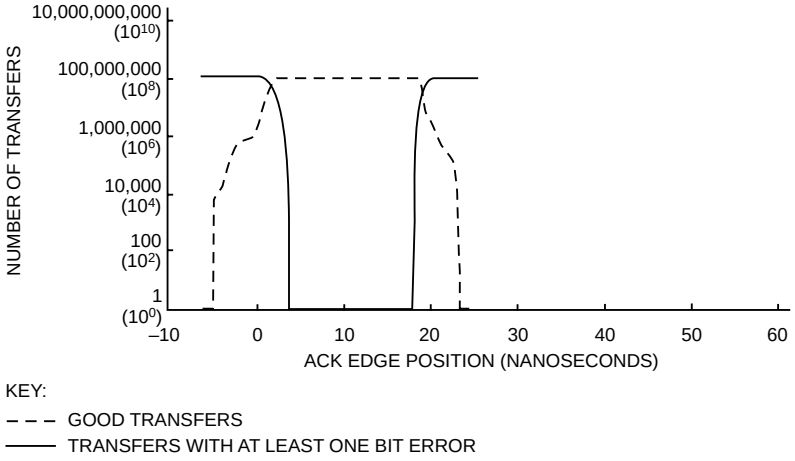


Figure 3
Fast-40 Error Rate Plot

Additional Tests Other tests that are useful with the special test environment are ground offset effects, terminator power effects, correlation of time domain plots on the signals with error rate distributions, hot-plugging testing (which results in good error detection), and comparison of the impact of different cables and transceivers. Test results of this nature are not included in this paper because the impact of these variations depends on many parameters and the results may not be generally applicable.

Timing Specification Methodology

With the increased emphasis on timing precision for UltraSCSI technology, it was necessary to introduce better specifications for the measurement of timing parameters than those in the SCSI-2 and SPI standards. Figure 4 shows the precise measurement points and features used for the specification of single-ended UltraSCSI signals.

The effects of the finite slew rate on the signal edges are accounted for largely by specifying the voltage levels that coincide with the receiver input levels. Thus, the setup time ends when the receiver is able to detect an asserted state at 1.3 V, and the asserted period begins when the asserted state has been detected. On the negation side, the signal must rise to at least 1.6 V before the receiver can detect a negated state, and a negated state must be detected if the input signal reaches 1.9 V. In the SCSI-2 and SPI standards, any point between 0.8 V and 2.0 V could be used as the timing measurement.

Sample UltraSCSI Signals

Numerous variations on the details of the signals can be produced in UltraSCSI configurations. This section shows two types of signals as representative examples that validate UltraSCSI as viable under certain conditions. The first case explores a configuration that actually exceeds the recommended specifications. This is a complex cabled environment with a cluster of loads on one end and some distributed loads on the other end. The second case shows the signals over a 25-meter, single-ended point-to-point bus.

Complex Loads Figure 5 specifies a complex configuration and the single-ended SCSI signals that result at various positions along the bus. The logic

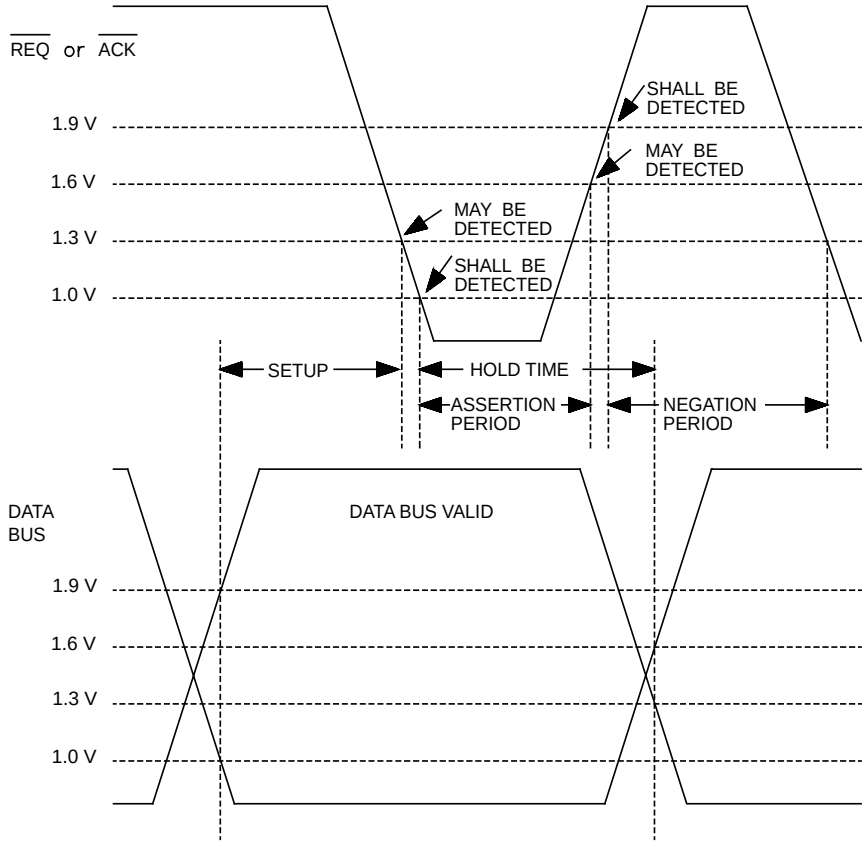


Figure 4
Single-ended UltraSCSI Timing Measurements

signal that is driving the SCSI driver chip is the first trace at the top; it provides a common timing reference for all the signals. The weakest signal is at device position 4, just after a relatively long run with no loads. This signal is below the 1-V level but has a very slow assertion slew rate that causes considerable loss of asserted state pulse width. This complex configuration works with the receivers used but does not have the timing margin required by the Fast-20 standard.

By varying the position of the loads so that there are no loads between the driver and the first load (not shown), the signal at the first load device is degraded even more than at position 4 in Figure 5. This is one reason that the overall length of single-ended UltraSCSI with many loads is restricted to 1.5 meters and that the total number of loads is limited to 8.[1] UltraSCSI devices connected to backplanes may be especially sensitive to attached cables that extend the total bus length more than 6 to 8 centimeters (cm) beyond the backplane. This reduced bus length is rather severe when compared to that allowed at the maximum fast SCSI transfer rate (a total of 3 meters).[4] In the section Small, Improved Interconnect, we show how to overcome this 1.5-meter, 8-device limit by using an active SCSI interconnect.

Applying the timing measurement methods shown in Figure 4 to the waveforms in Figure 5 illustrates that more careful timing specification methods do indeed help significantly to keep the timing margin high enough to use.

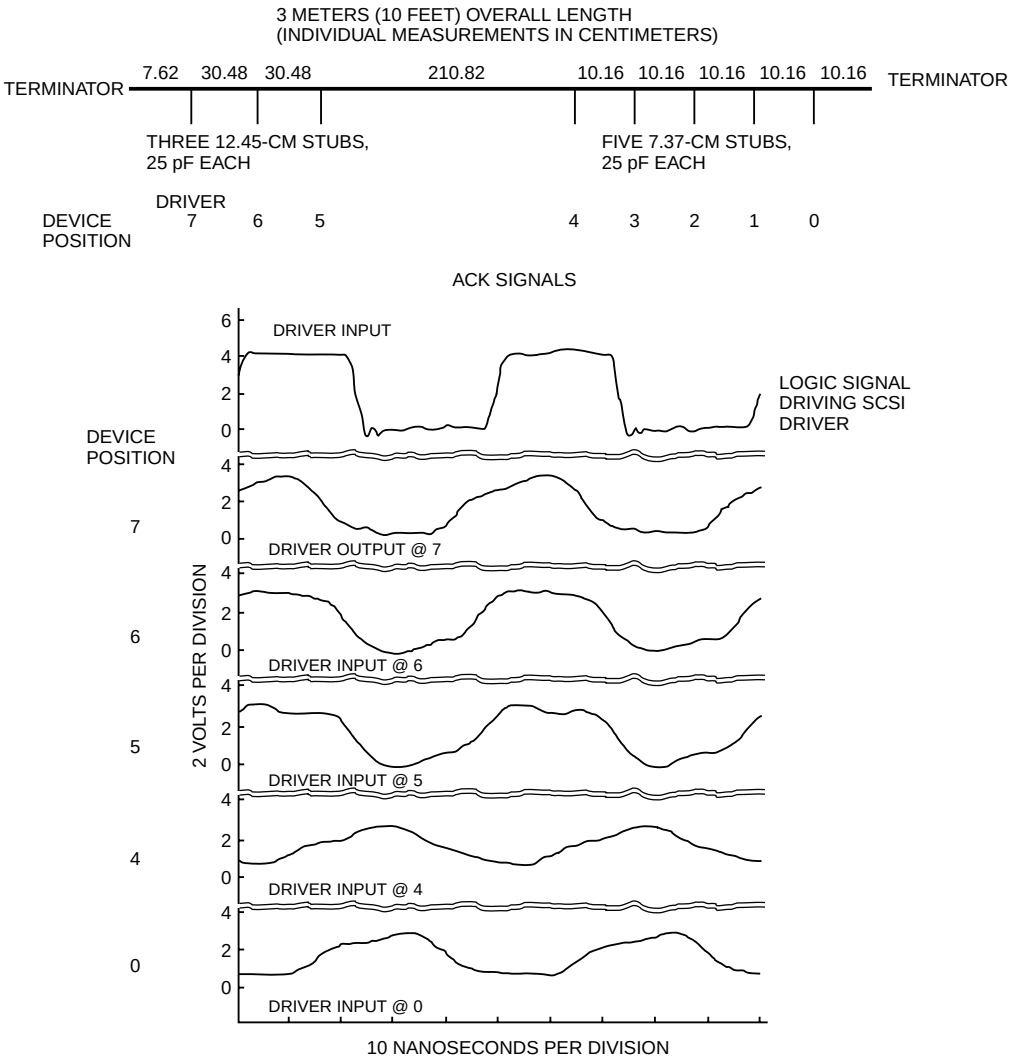


Figure 5
UltraSCSI Signals in a Complex Bus

Point-to-point Configuration If loads are present only at the ends of the bus, the transmission line between SCSI devices improves electrically. This occurs simply because the loads significantly disrupt the characteristic impedance and cause reflections and attenuation. The point-to-point signal at 25 meters has better amplitude and timing margins than signals in much shorter buses with closely spaced loads. Figure 6 shows a typical example of a point-to-point UltraSCSI signal. The format used in Figure 6 is the same format used in Figure 5.

Differential UltraSCSI

Differential UltraSCSI uses the same configuration rules as fast SCSI (25-meter total length, 20-cm [8-inch] stubs, 16-device load)[1] and uses the same timing values as single-ended UltraSCSI. The larger signal amplitudes and the common mode rejection property of differential transmissions help overcome the transmission line weaknesses in heavily loaded and long buses. As with any high-voltage differential system the costs—in terms of money, power, and space—are higher.

Other Requirements for UltraSCSI

The Fast-20 standard[1] contains a number of detailed requirements on the components used in UltraSCSI configurations. Included are slight modifications to the cable impedance, active negation requirements for drivers, special length limits for certain loading conditions, restrictions regarding the kinds of single-ended terminators to use, and timing budgets.

Summary of Developments in the Area of Increased Synchronous Data Phase Speed

The UltraSCSI (Fast-20) speed increase can be attributed to a systematic examination of the margins present in actual SCSI hardware and to the elimination of the excess margins. Advances in the integrated circuit industry enabled silicon designs to be specified with tighter controls on the driver and receiver timing and threshold properties than were possible when the SCSI-2 or SPI standards were developed. All the important changes needed for SCSI devices are contained in the silicon designs for the drivers and receivers. As a result, the user sees no difference between the appearance of UltraSCSI and that of ordinary SCSI.

The system integrator must use a more restrictive set of configuration rules than required for fast and slow SCSI. Also, the only impact on software is the addition of a new speed agreement code for the rates uniquely supported by UltraSCSI. This negotiation is done precisely the same way for UltraSCSI as for any other form of SCSI. Finally, UltraSCSI devices are 100 percent backward compatible with fast and slow devices. Although a device may be capable of the maximum UltraSCSI rate, it may be needed in a configuration that does not support UltraSCSI. In such a case, the UltraSCSI device would be used in the fast or slow mode and would have more margin at those slower speeds than it would if it were not UltraSCSI capable.

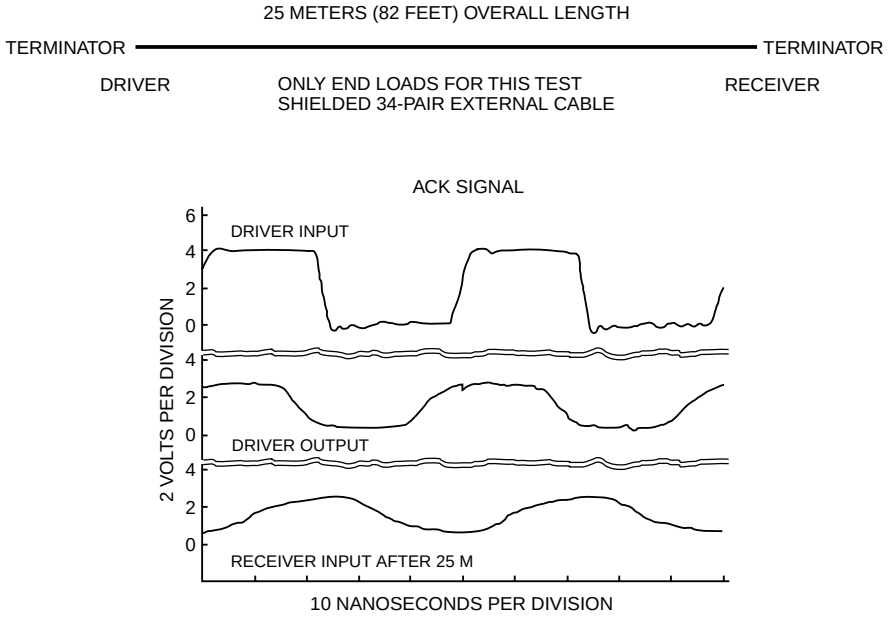


Figure 6
Point-to-point UltraSCSI Signals

Bus Expanders

As noted previously in the discussion of complex loads, there are rather severe limits on the configurations that can be achieved with single-ended UltraSCSI when implemented in a single bus. The extension to parallel SCSI architecture that overcomes this constraint involves using active circuits that connect SCSI buses electrically but isolating them from each other in a transmission line sense. These circuits have the general name expanders, since they expand the configuration capabilities of parallel SCSI.

Each individual bus has two terminators and its own transmission mode (single ended or differential) and obeys transmission line–based configuration rules as if it were the only bus in the system. When used with expanders, these individual buses are called bus segments. The collection of SCSI devices in all the bus segments that are electrically connected together is called the SCSI domain. One example of a SCSI domain using expanders is shown in Figure 7. Note that when using expanders, it is possible to have bus segments that do not have any SCSI initiators or targets but only serve to form an electrical interconnect between other bus segments.

Expander Properties

Expanders are available in two basic types: simple and bridging. Bridging expanders behave as a SCSI initiator or target, whereas simple expanders have a set of properties that make them look like a piece of wire with delay to the protocol. Simple expanders

- Cannot initiate SCSI IDs and arbitrations and cannot originate messages, although the expanders can read messages sent from initiators and targets
- Allow minimal arbitration propagation delay
- Yield a retransmitted signal timing skew (both delay and high/low) no worse than from valid SCSI initiators or targets
- Do not interfere with the REQ/ACK offset count
- Allow min/max pulse widths to be maintained
- Require the filtering of the SCSI RESET line
- Allow arbitrary placement of the initiator and the targets
- Require that terminator power not be connected between the segments being coupled
- Do not need to know the negotiated data phase speed or any other variable property of a transaction
- Require that there be no electrical or logical connection of the DIFFSENS line (a single-ended signal that indicates the transmission mode being used on the bus segment) between segments being coupled
- Issue a SCSI bus RESET signal on one segment on detecting transmission mode (single-ended/LVD, etc.) changes on the other segment

Simple expanders are becoming available from several sources in the industry for use with UltraSCSI.

Domain Rules Using Simple Expanders

When using only simple expanders in a domain, six rules must be observed:

1. All bus segments in the domain must comply with their individual bus segment length limits and other segment-related requirements.
2. Any segment between two other segments must support the highest performance level that can be negotiated between the two other segments. For example, two wide UltraSCSI segments must not be separated by a segment that does not support both wide SCSI and UltraSCSI.
3. The maximum propagation delay between any two devices in the domain cannot exceed 400 ns. A special case exists for devices that use extremely long times for responding to BUS FREE (the so-called BUS SET DELAY)—the one-way propagation limit is 300 ns instead of 400 ns.
4. The number of addressable devices cannot exceed 16 unless the domain contains bridging expanders.
5. A branch/leaf architecture must be observed; loops are not allowed.
6. The REQ/ACK offset negotiated between any two devices must be large enough to ensure that adequate offset and buffering is available to accommodate the round-trip time between the devices. For the maximum UltraSCSI rate with a 400-ns maximum one-way domain propagation time, the

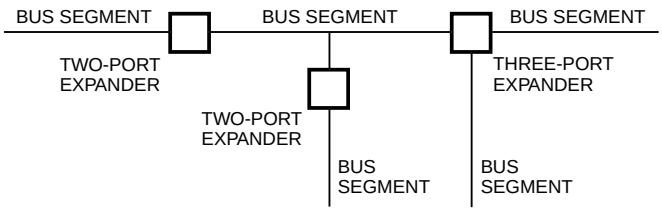


Figure 7
SCSI Domain Built Using Expanders

minimum offset is 18. (This offset level is derived by considering a maximum round-trip time of 800 ns at 50 ns per transfer [800/50 = 16] and somewhat arbitrarily adding two transfers to account for some additional delay due to the processing time in the silicon.)

Achieving the 400-ns one-way domain delay requires expanders that will not pass the wired-OR glitch (noted earlier in the introduction) between bus segments. This filtering of the glitch allows the bus segments to settle individually.

The propagation delay through an expander directly subtracts from the physical distance between devices. It is therefore desirable to use expanders with small delays. For a single-ended–to–single-ended application, the delay can be as low as 10 ns. For a single-ended–to–differential application, the delay is typically around 100 ns, which is another significant penalty to using differential bus segments.

More detail concerning these rules and other properties is available in the draft ANSI document: *SCSI Enhanced Parallel Interface,*[5] which was edited by the author of this paper.

Summary of Improvements Related to Bus Expanders

The use of simple expanders dramatically extends the utility of single-ended UltraSCSI. The most obvious example is the ability to introduce point-to-point segments where additional length is needed. A less obvious example is the ability to create star or hub configurations by clustering simple expanders into a local physical area. An example of a three-port SCSI hub is shown in Figure 7. Note the three simple expander circuits internally connected within the hub. Simple expanders also make it possible to mix single-ended and differential SCSI devices in the same domain, to achieve the full 16-device count, to add and remove bus segments without shutting down the entire domain, and to achieve differential performance without incurring the extra cost of differential. Bridging expanders offer the same transmission isolation as simple expanders and may allow increasing the number of devices in the domain to as high as 946,[5] but bridging expanders are not as well developed as simple expanders and will not be explored in depth in this paper.

Note that the improvement in signal integrity is dramatic when using expanders with backplane applications. Therefore, it is good practice to use an expander whenever connecting a SCSI cable to a backplane that contains SCSI devices.

Smaller, Improved Interconnect

Another recent development in parallel SCSI technology is the introduction of much smaller external physical interconnects and more capable internal device interconnects. The SCSI connectors and shielded cables have historically been large, bulky, and generally difficult to manage.

Spearheaded by activities that began in 1995 in the SFF (formerly Small Form Factor) industry group, standardization is under way of two new connector families that offer unprecedented levels of functionality and true multisourcing of complete connectors for parallel SCSI. These families are the Very High Density Cabled Interconnect (VHDCI)[2] shielded connectors that reduce the overall size of an external connector by two thirds and the Single Connector Attachment–2 (SCA-2)[7] unshielded connectors that integrate into a single connector all the functions needed to run a peripheral. The VHDCI family revolutionizes the external SCSI interconnect and the controller parts of the internal SCSI interconnect; the SCA-2 family does the same for the internal device interface.

For the first time, complete connectors—not just the mating interface—are being standardized. This feature is essential to achieving interchangeability and second sourcing for connectors with the same style of termination-side contact. The VHDCI family is specified in 26 different forms, all with exactly the same mating interface, so that virtually any kind of device or cable assembly design can be accommodated. Interestingly, this array of choices for the connectors does not increase the complexity of the interconnect but rather opens up new ways for product developers to design products while maintaining a simple and physically interoperable separable connector interface. In fact, this ability to accommodate a variety of product design requirements without changing the separable interface is one reason that SCSI is becoming *less* complicated.

Similarly, the family of SCA-2 connectors for SCSI internal devices and cables is following the VHDCI standardization model, with a significant number of intermatable forms being standardized. These connectors offer the ability to bring all the SCSI signals, all the power and ground connections, and all the optional signals, such as IDs, spindle sync, and power fail, out of the device through a single unshielded connector. This feature dramatically shrinks the cost and complexity of interconnecting an array of SCSI devices.

Using an SCA-2 connector, the device may be inserted into a backplane without using cables. If the SCA-2 and backplane combination is not used, a SCSI cable (50-pin or 68-pin conductor), a four-lead power cable for ground and power (5-V and 12-V), and one or more smaller cables for the IDs etc., are required for *every* device in the system. Each of these cables is routed differently, has different current carrying and other electrical requirements, and has very different connectors. Although this cabled option is flexible and offers significant advantages in some systems, it is usually not the best solution in the device array and modular packaging applications that are required for the

higher-end applications. Therefore, the SCA-2 is a significant factor in the dramatic reduction in complexity of higher-end SCSI device applications.

VHDCI Connectors

The physical size of the VHDCI connectors is much smaller than the earlier versions, as seen in Figure 8. Because of its low profile, the VHDCI 68-pin family is approximately half the height and twice the width of the latest Fibre Channel external connector, the High-Speed Serial Data Connector (HSSDC). Figure 9 shows a comparison of the VHDCI and HSSDC connectors. The same panel space is required for either technology.

The VHDCI connectors shown in Figure 9 are closely spaced, but the orientation of the polarizing shield connection is 180 degrees different between the upper and lower connectors. This arrangement allows an offset cable assembly to be used where one side is flat. This same cable assembly may be used on both the upper and lower connectors without interference. The specifications of the VHDCI interface ensure that neighboring PC option slots will not have interference even if all the SCSI ports have cable assemblies attached.

The VHDCI connector is useful for multiport applications such as RAID (redundant array of inexpensive disks) controllers. Figure 10 shows examples in which the wide version of the connector family has allowed at least a doubling of the number of ports possible in a single controller form factor. As illustrated in Figure 10, the device design enables up to four wide SCSI ports on a single PC option card cutout.

The VHDCI retention scheme is also significantly simplified by introducing a three-way retention post for the bulkhead connector. This post accepts (1) the conventional jackscrews, (2) a squeeze-to-release clip for positive retention with rapid release, or (3) a detent ring retention that requires a stronger pull than that required with no retention but no action other than

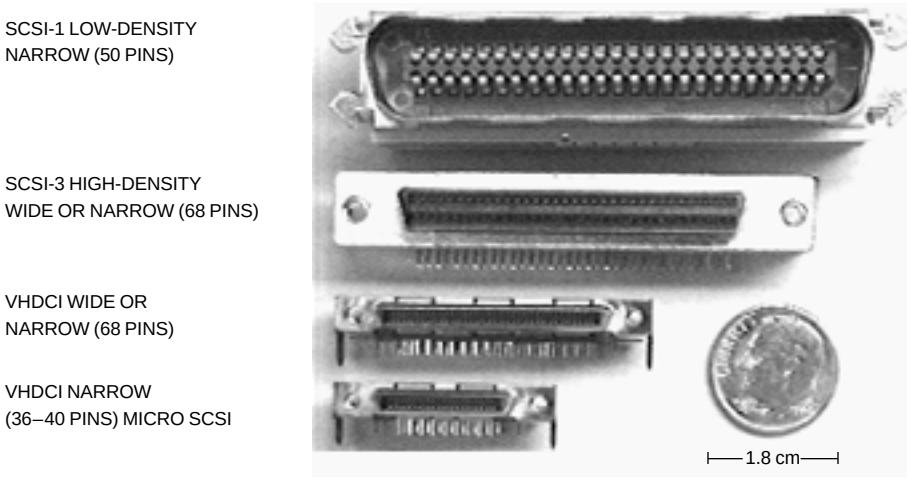


Figure 8
External SCSI Connectors

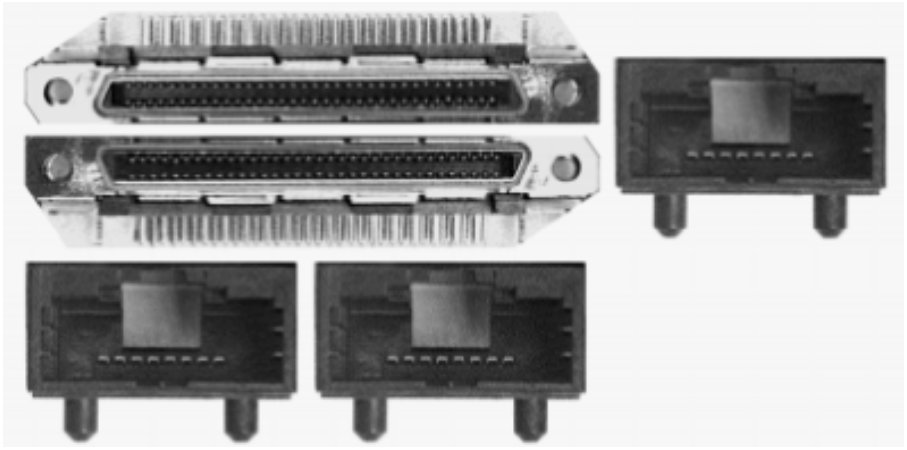


Figure 9
Comparison of the 68-Pin VHDCI and Fibre Channel HSSDC Connectors

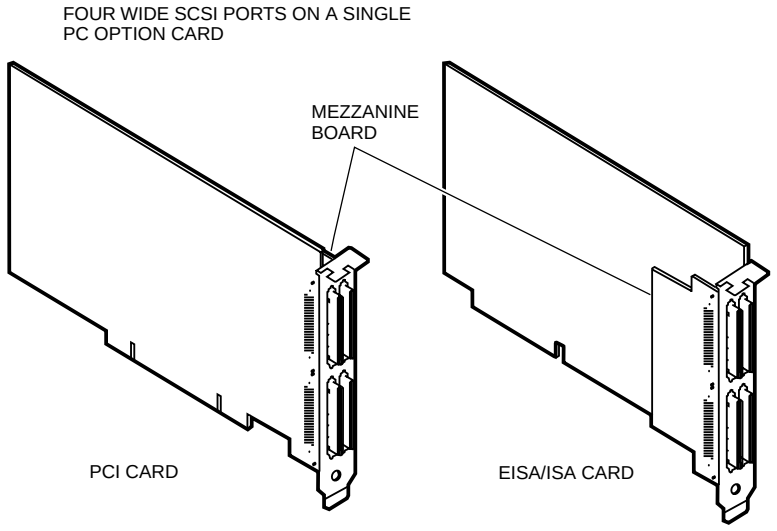


Figure 10
Four Wide SCSI Ports on a Single PC Option Card

pulling or pushing. The choice of retention type is made in the cable assembly. All 68-pin VHDCI cable assemblies that comply with the SFF specifications work on all 68-pin VHDCI mating connectors.

Figure 11 shows the details of the 68-pin VHDCI system. The lip in the jack post provides the securing point for squeeze-to-release clips and for split-ring detent retention. The center of the jack post is threaded for use with jackscrews.

Although smaller than the high-density connector, the VHDCI connector is durable. It has no pins that can bend; its retention scheme uses the same-size jackscrew thread as the high-density wide connector; and its contacts are imbedded in the housing where they cannot move or become distorted.

SCA-2 Connectors

The SCA family uses an 80-position, leaf-style contact to interface all active SCSI lines, three power voltages, and device control signals. This connector is considerably smaller than the collection of the three different connectors used for power, options, and SCSI bus in a cabled system. There are two basic versions of SCA connectors: SCA-1 and SCA-2. Both versions are unshielded and useful only within shielded enclosures. The SCA-1 has 80 positions with all contacts designed

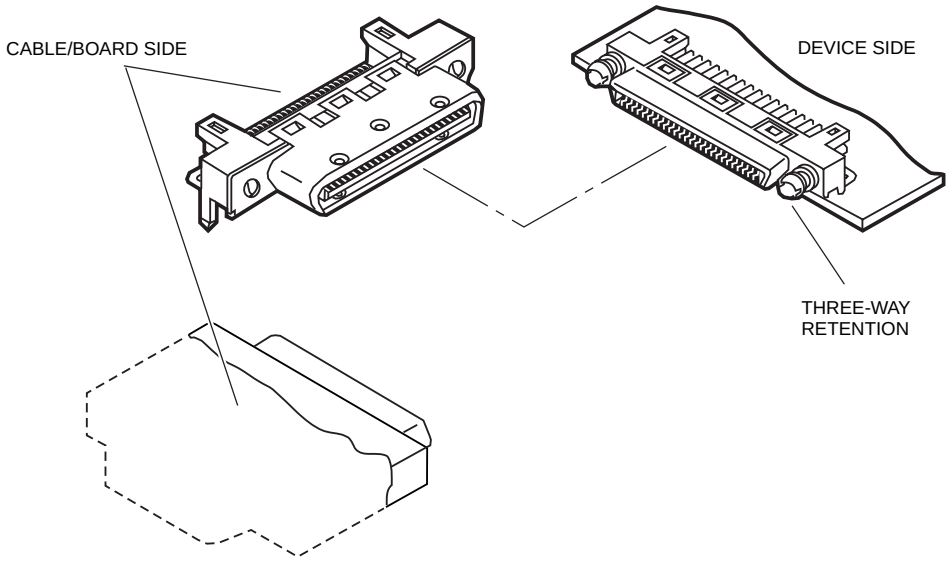


Figure 11
Overall View of the 68-pin VHDCI System

to be the same length. The SCA-2 can be mated to the SCA-1 but has advanced grounding contacts and sequenced signal and power contacts for supporting hot plugging and blind mating (no visual feedback during mating). Both versions are available in many styles, which differ by the termination-side structure and overall orientations.

The SCA-1 is not a documented standard and is being replaced by the SCA-2. The SCA-2 connector was introduced to SFF in 1995[7] as the first step toward formal standardization.

Two special features exist in the SCA-2 connector. First, two contacts, one on each side of the connector, provide the first make/last break for the ground connection. This design ensures that a common electrical ground is established between the device and the system before any power or signal connections are made on device insertion. Upon removal, these contacts ensure that the ground stays intact throughout the disengagement of the signal and power pins.

The second feature allows the special long power contacts to precharge bypass capacitors before the main power contacts make. This reduces the disturbance to the power distribution system and eliminates any arcing on the service power pins. Two pins at the extreme ends of the connector indicate that the connector is fully mated. The overall view of the SCA-2 system is shown in Figure 12.

The size of the connectors in the SCA family has not decreased dramatically. The connectors need to maintain enough size to achieve blind mating alignment, and, for backplane applications, there is little advantage in having a connector that is smaller than the device. With 89-mm (3.5-inch [in]) or the newly proposed 76-mm (3-in) form factor devices, the SCA connector comfortably fits within the device boundaries.

The use of backplanes for direct device attachment is possible because all the electrical connections for the device are available in one connector on the device. This design eliminates the cables used to attach the device and the space required for the connectors, thus significantly shrinking the size required to package multiple devices.

External SCSI Cable

The external cable for SCSI is shrinking also, through the use of smaller-gauge wire, better dielectrics, and less jacketing material, as illustrated in Figure 13. Formerly, wide SCSI required a cable of approximately 12.70 mm (0.50 in) in diameter (a 126.677-mm^2 [0.196-in^2] cross section) with 28-gauge wire. Today, wide SCSI cables with 30-gauge wire are shipping with diameters of 9.40 mm (0.37 in) (69.398-mm^2 [0.107-in^2] cross sections). Cables with 7.62-mm (0.30-in) diameters (45.61-mm^2 [0.07-in^2] cross sections) are possible with 32-gauge wire and inexpensive dielectrics for wide SCSI. Cables with 6.35-mm (0.25-in) diameters (4.987-mm^2 [0.049-in^2] cross sections) for narrow SCSI (45.61-mm^2 [0.07-in^2] cross sections) are flexible and manageable—similar in size and flexibility to a desktop computer power cord and smaller than many serial cables. When used with active single-ended, LVD, or HVD terminators, the 32-gauge wire is adequate for distributing terminator power and SCSI signals in most applications. Long cables should not be used for terminator power distribution.

Further reductions in the connector and cable sizes need to be weighed against the ease of handling, the need for sufficient strength to survive normal service stresses, and the cost increases at very small sizes. The combination of the VHDCI connector and 30/32-gauge wire sizes is a good optimization.

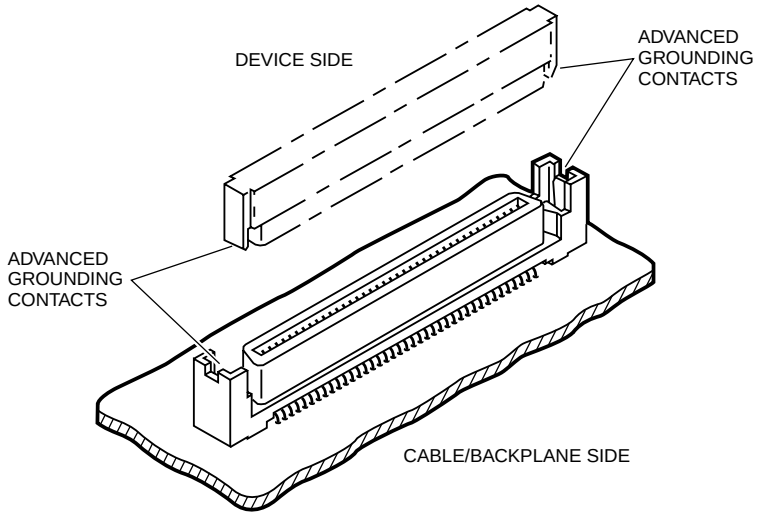


Figure 12
Overall View of the SCA-2 Connector System

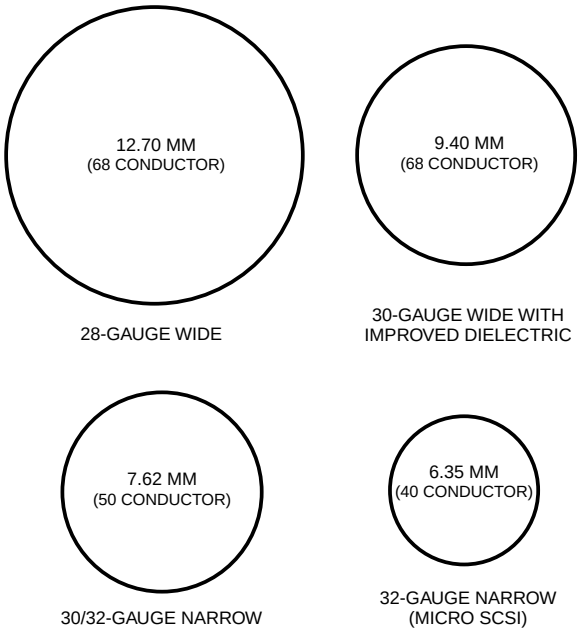


Figure 13
External SCSI Cable Diameters

Summary of the Benefits Derived from a Smaller, Improved Interconnect

The VHDCI connector and smaller cables combine to offer a robust yet user-friendly revolution in SCSI interconnect. The leaf-style contact of the SCA connector eliminates problems with bent pins that frequently bedevil the older wide SCSI connector. The ability to use up to four wide UltraSCSI ports in a single PCI option slot increases the SCSI connectivity per PCI slot to 60 devices (from 15 devices). By using multiple PCI slots, hundreds of SCSI devices can be connected to a single PC or workstation. In addition, the SCA-2 connector implements the essential contact sequencing required to perform SCSI device hot plugging.

Device Insertion and Removal Bus Transients

The multidrop feature of the SCSI bus allows device removal and replacement without disturbing the communications between other SCSI devices, if the electrical disturbances caused by the device being added or removed are not detected by any other SCSI devices. Thus, it is architecturally possible to dynamically reconfigure the device population without interrupting existing data transmission processes between operational devices.

The transients involved with device insertion and removal include mechanical vibrations, power distribution instabilities, SCSI terminator power noise, electrostatic discharge (radiation and induced current), and SCSI signal line noise. All except the SCSI signal line noise and the terminator power noise are handled by the storage system design and therefore are not directly part of the advancements in parallel SCSI. The SCSI terminator power noise is determined by the size of the decoupling used on the SCSI terminators and the size of the capacitance on the device being inserted. This noise is easily controlled by ensuring that these sizes meet the values specified in the SPI standard.[4]

The delicate case is when the SCSI signal lines are involved, which is the subject of this section. To determine the nature and magnitude of these signal line disturbances, one must understand the following three mechanisms: (1) the overall sequence of events, (2) the electrical dynamics of connector contacts when used in the SCSI application, and (3) the electrical consequences on the bus when the device makes/breaks contact with the SCSI signal line.

There are two sequences of interest: insertion and removal. The removal process is easy to grasp after the insertion process is understood.

Single-ended Device Insertion

The initial conditions considered for SCSI device insertion assume a SCSI device with its ground solidly and continuously connected to the ground of the SCSI bus. This connection is easily accomplished, for example, by using sequenced contacts where the device ground makes connection well before any signal connection. In this state, the SCSI device pins present a maximum fully discharged capacitance of approximately 25 picofarads (pF). After the device signal pin contacts the bus, this capacitance becomes charged (by extracting charge from the bus) to the voltage on the signal line at the time of the insertion.

These values range from approximately 3 V for negated lines to nearly 0 V for asserted lines.

Since the SCSI device being inserted is logically off (i.e., there is no driver current), the only current that needs to flow is that required to charge the 25-pF capacitance. This is sharply different from many connections in electronics in which current flows through the contact after an electrical contact has been established.

In the case where no bus voltage changes occur except as a result of the device insertion, the insertion transient begins with the initial contact and ends when there is no further bus voltage change with time (the steady state voltage). Once the device pin voltage reaches the steady state bus voltage, no further current flows through the contact.

Therefore, once the device capacitance becomes charged to the steady state signal line voltage, no further disturbances to the signal line voltage will occur even if the contact opens momentarily during a chattering event. The voltage on the device capacitance changes during the transient from a discharged state (zero voltage) to the steady state signal line voltage, with the current always flowing into the device capacitance.

If the signal line voltage changes after the insertion transient is completed (because of events such as being driven by other devices, by noise, or by the inserting device beginning to use its own driver), then current will again begin to flow through the contact. This is a normal SCSI condition for contacts in service. If the signal line voltage changes during the insertion transient because of events other than the connector contact effects (e.g., signals changing because of being driven by other devices, other noise), then it is more difficult to determine exactly where the insertion transient ends. The beginning of the insertion transient will still be marked by a charging of the device capacitance. Examples presented later in this paper show both insertion events and driving events from other devices occurring at the same time.

The time required for complete contact mating on all SCSI signals in the bus is up to six orders of magnitude greater than the time required for a SCSI signal to change state. Therefore, signal level changes are likely during the insertion process. The electrical behavior of the contact as it continues wiping (sliding after initial contact is made) from its initial contact point to its final resting position becomes a critical part of the process. The following subsections explore this behavior in detail.

Connector Insertion Dynamics The data presented in this section were derived from a DIGITAL DSSI bus in 1990. The DSSI bus is nearly identical to the SCSI bus, and many of the results apply without modification to SCSI. Similar data have been observed on the SCSI bus, but the complete set of data presented in this paper is not presently available from actual SCSI hardware. The disturbances in the DSSI bus are larger than those seen in the SCSI bus, because the DSSI voltages are slightly higher (3.5 V for DSSI compared to 2.8 V for single-ended SCSI), and the instrumentation capacitance (~10 pF) adds significantly to the device capacitance because of the state of the art for scope probing in 1990. Numerous tests with modern scope probes (0.6 pF or less) of SCSI hardware have shown that the SCSI disturbances are indeed qualitatively the same but significantly less in size than those shown here from the DSSI hardware.

The mechanisms described apply to any system in which the insertion transient is caused by the charging of a small capacitance. Figure 14 shows the basic test setup. A device is inserted into a connector with scope probes attached on either side of the mating interface and with an additional probe attached to the bus some distance from the connector. The voltage on the device side of the connector is used as the trigger signal into a digital storage scope so that the events before, during, and after the mating event can be examined. This is clearly a single-event type of measurement, so a high sampling rate (1 billion samples per second) and significant scope memory is required to capture the waveforms. The scope probes used have a 1-megohm input resistance.

The connector used for the tests in this section has multiple parallel pins that all mate and demate in the same general time period. There is no intentional difference in the pin lengths. The time relationship between the mating events on two neighboring pins was explored first. By choosing neighboring pins, the differences between the pins is kept to a minimum so the time differences observed should represent the best pin-to-pin synchronization in a mating event.

For this test, a probe was attached to each of two pins, and the connector alone (not part of a device) was mated to the bus segment connector. Figure 15 shows the results.

Both pins appear to show instantaneous transitions between the charged and discharged states on the time scale that was required to capture both events on the same plot. The mating events are separated by approximately 19 milliseconds, and there is no evidence of any discharging after the initial charging has occurred. Since the scope probes have a 1-megohm input resistance, any lack of contact during the wipe portion of the mating will allow the capacitance to discharge through the probe with a time constant of approximately RC, where R is the scope probe resistance and C is the sum of the connector pin and probe capacitances. Assuming a total of 10 pF, this gives a decay time constant of 10 microseconds.

Figure 16 shows another mating event on pin 1 at a 500 times more sensitive time scale. In this case, some evidence of momentary opens is seen with the

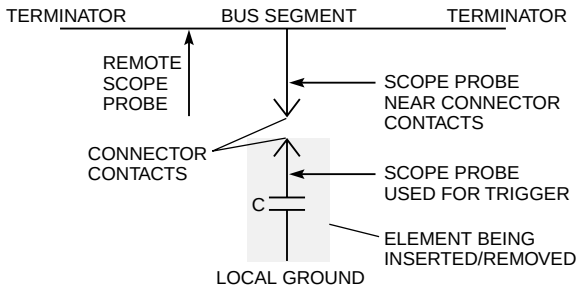


Figure 14
Test Setup for Insertion/Removal Transients

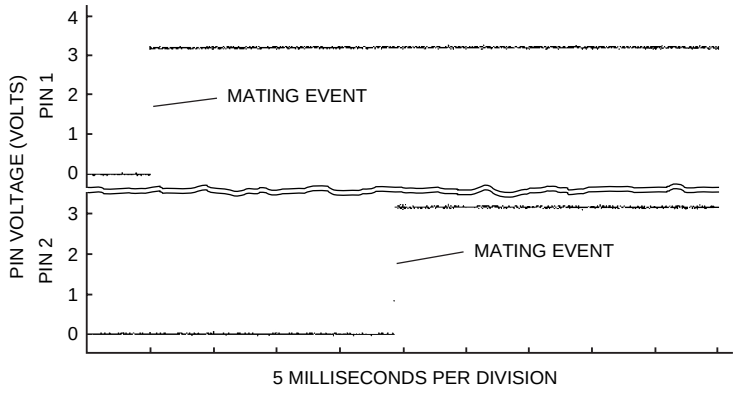


Figure 15
Time Relationship between Mating Events for Two Pins in the Same Connector

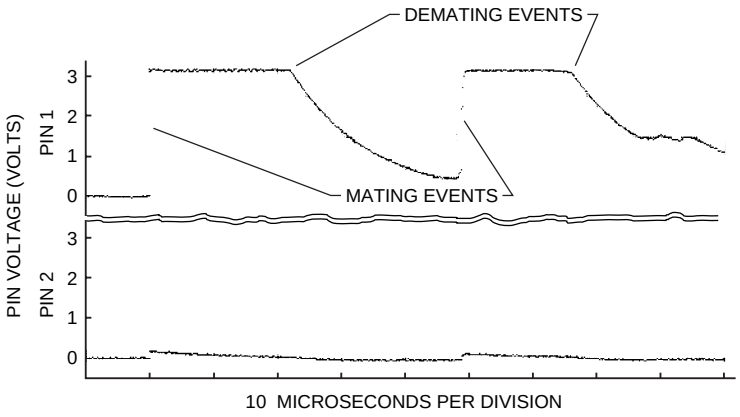


Figure 16
Contact Bounce Events

expected decay dynamics. The actual time constant is a bit longer than 10 microseconds because of some capacitance in the connector pin. This bounce behavior may or may not be present during the initial stages of the event shown in Figure 15, but clearly the behavior is not visible in the figure. To observe the suite of transients that exist in the mating process, one must examine the transients at several different time scales. In general, this requires repeating the mating events, since the dynamic range of the scopes used was insufficient to capture all the detail in a single event.

The initial mating event on pin 1 still appears to be instantaneous on the time scale used in Figure 16, but some slope is visible in the second bounce event. Also, during the second decay period, a shelf in the decay indicates that a partial, high-resistance contact was briefly experienced. Pin 2 is not close to making a contact at the time range shown in Figure 16. The figure shows a small amount of cross talk in the pin 2 voltage waveform caused by the pin 1 transients.

This data clearly shows that the details of the mating process are highly complicated and intrinsically

unpredictable. Therefore, the best we can hope for is to establish some limiting cases for the important parameters. The limiting features shown in Figure 16 are the extremely rapid initial mating event and the decay times. We examine these rapid transients in detail later in this section. The decay times are determined by the actual contact resistance and the resistance of the leakage path to local ground. For normal SCSI devices, there is very little leakage to ground on the device pin so the opens produced by the bounce have no effect.

Some cases observed indicate much more complex bounce structures. Figure 17 shows a case in which the mating connection is not established until more than 700 microseconds have passed.

The data in Figures 15 through 17 were all acquired from the same connector contact during separate mating processes. Typically, the details of the mating event are very different even under nominally identical conditions.

Another type of mating event is shown in Figure 18. This event requires approximately 10 microseconds to make the transition from uncharged to charged, and there is no bounce. This particular event produces almost no cross talk into pin 2. Events with these characteristics are somewhat rare and are called gradual transients in this paper.

Figure 19 shows a closer look at the rapid transient type of mating event. In this figure, we have added a device capacitance of approximately 20 pF to the scope probe for a total of approximately 30 pF. Notice that the transient requires 2 to 3 ns to substantially complete its charging. There is a ratio of nearly 10^7 between the mating events on different pins in the same connector and the rapid transient of a single contact.

Limiting Parameters for the Rapid Transient The question of whether the rapid transient shown in Figure 19 is the worst case needs to be explored because the duration of the transient affects the disturbance on the bus. Some bounding features and some implications of the observed behavior of the rapid transient are noted in this subsection.

Assuming that the transient event occurs in 2 ns and that the velocity of impingement just prior to the first mating event is 1 meter per second, then the distance traveled by the contact would be 2 nanometers (nm).

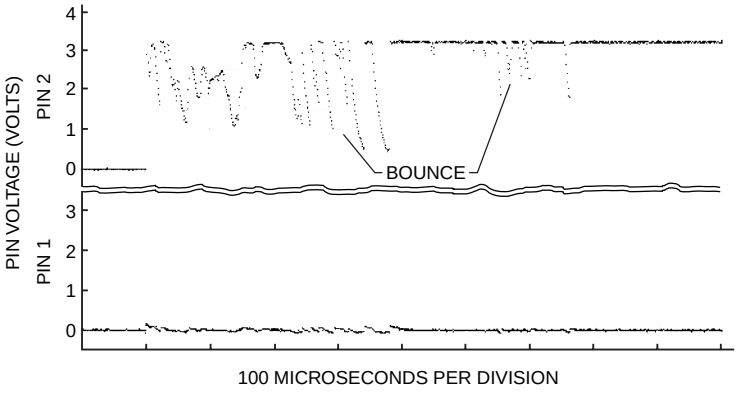


Figure 17
Extended Mating Bounce Events

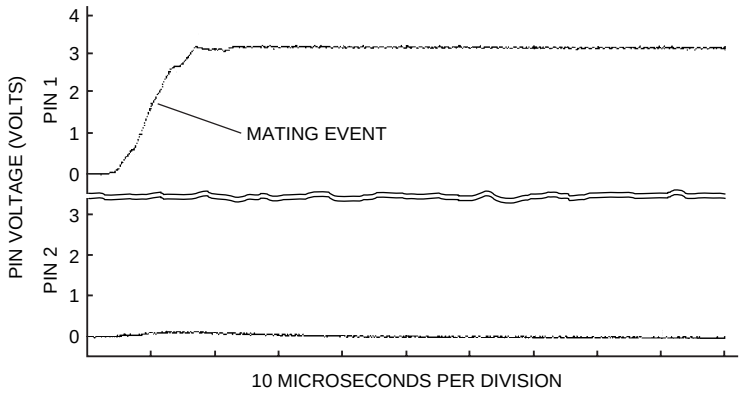


Figure 18
Gradual Mating Event, No Bounce

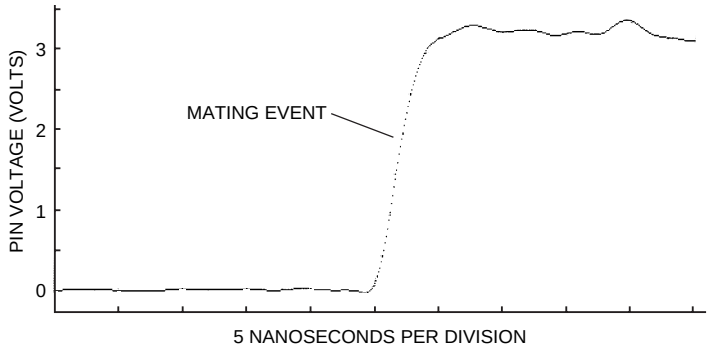


Figure 19
Detailed Structure of the Rapid Transient

This distance is equivalent to a few atomic distances. The distance traveled during the gradual transient shown in Figure 18 would be approximately 10 microns, and during the extended bouncy case shown in Figure 17, approximately 1 mm. The velocity for the latter two cases would likely be somewhat reduced because of the mechanical interference between the pins, and the actual distance traveled is probably significantly less. There is little opportunity, however, for the velocity to be reduced for the rapid transient, and this distance of 2 nm is probably at least the correct order of magnitude.

The following calculation shows the total current levels required to charge the capacitance in 2 ns.

$$Q = CV = 30 \times 10^{-12} \text{ pF} \times 3.5 \text{ V}$$
$$= 10.5 \times 10^{-11} \text{ coulombs},$$

where Q represents the total charge, C is the capacitance, and V is the voltage. Since this charge is transferred in a time t of 2 ns, the average current is

$$Q/t = 10.5 \times 10^{-11} \text{ coulombs}/(2 \times 10^{-9} \text{ ns})$$
$$= 52.5 \text{ milliamperes (mA)}.$$

For a gradual transient that takes 10 microseconds, the average current is approximately 10 microamps. These calculations show that the most severe amplitude disruption to the signal on the bus occurs with the rapid transients, since relatively large current must be supplied in a short time to charge the capacitor.

The next item to be examined is the current density that must exist during the transient. Since the contacts move only 2 nm and the surface finish of actual contacts is not nearly this smooth, it is reasonable to assume a square 2-nm contact. Clearly, this assumption is not rigorously defensible and could be the subject of an entire study area in its own right; however, there is no basis for assuming that the lateral contact region would be any different than the contact area in the mating direction. The basic conclusions would not be affected even if we assumed a hundredfold lateral increase in contact area. Attempts to use scanning electron microscopy to examine the actual contact area were not fruitful in establishing the actual initial physical contact area because of the severe physical disruption that occurs on the microscopic level and because of the small sizes involved.

Under these assumptions, the physical contact area is assumed to be $(2 \text{ nm})^2$ or 4×10^{-14} cm^2 in the following calculations. The current density to support the 50-mA rapid transient current is therefore approximately 10^{12} A/cm^2. Typical current densities in copper and other metals are less than 10^6 A/cm^2. The electromigration onset current is of this same order. The current density in the rapid SCSI transient is a million times greater than that which metal can support.

To support the massive current density, the actual contact area must be much larger than the initial physical contact area assumed in the above calculations. The author believes that this can be explained by a micromolten metal-to-metal joint that is formed upon initial contact and that the front of the melt propagates (probably through phonon interaction) at approximately the speed of sound in the metal. This process would create crudely a thousandfold increase in the effective insertion velocity and would result in a millionfold increase in contact area, since the melt would propagate in all directions.

This mechanism would produce reasonable current densities and would form an intimate metal-to-metal interface with both contacts that would aid in reducing the contact resistance. The micromelt size becomes rapidly self-limiting, with the expanding contact area causing decreased current density, which in turn, causes decreased melt temperature.

As discussed in the next section, the actual contact resistance during the rapid transient cannot be large. If this resistance is large, as in the case of the gradual transient, the mating event is much less disruptive.

Many variations on the mating transients can be observed, but we do not attempt to show all of them

in this paper. One special variation, however, is worth noting—the combination of rapid and gradual transients in the same mating process. Sometimes the mating process starts with a gradual transient and then shifts to a rapid transient. Figure 20 shows a complex mating process in which (1) a gradual transient initiates, (2) a rapid transient starts but does not complete, (3) the rapid transient ends, (4) another gradual transient process starts, and (5) another rapid transient finishes the charging process.

This observation is consistent with several possible microprocesses during which the initial rapid transient extinguishes before completion.

- The micromelt becomes physically torn apart by the advancing motion of the contacts. (This process is unlikely because of the excessively slow physical motion.)
- The micromelt explodes. (This process is likely.)
- The micromelt becomes resistive through the contamination of the melt with insulating material.
- The micromelt front reaches a thin region and opens because of the lack of material.
- The micromelt front reaches an insulating region.

On further movement of the contacts, a new rapid transient condition is encountered between different metallic peaks of the contacts, and a new rapid transient begins. Figure 21 shows a conceptual representation of this process.

Gradual transients appear to be associated with normal current densities (i.e., 10^6 A/cm^2) and much higher contact resistance than rapid transients. In cases where a micromelt is not initiated, the low contact resistance associated with the liquid metal–to–solid metal interface and the expanded contact area are not present. Therefore, one way to eliminate the mating disturbance caused by the rapid transients is to ensure that a micromelting process is not possible.

In the process shown in Figure 20, it is probable that a gradual-type contact is being maintained somewhere else in the contact, since no voltage decay is evident when the rapid transient ends. Indeed, it is to be expected that the rapid transient mechanism would not operate after the capacitance is charged to a certain level, since there would not be enough energy difference to initiate and sustain a rapid transient. Therefore, the gradual transient is the behavior derived from an extrapolation of the normal mechanisms that produce contact resistance. This detailed discussion is pursued because we must understand the basic physical mechanisms to gain confidence that we are considering the worst-case disturbances.

Single-ended Device Removal

During the process of removal, the device pin separates from the bus. Since both the bus and the device are at the same voltage just before the separation, no current is flowing unless the bus voltage changes when the contact is in the process of separating. Therefore, in most cases the separation process causes no disturbance.

Bounce can occasionally be observed during the demating process when there is a leakage-to-ground path present on the device side. Of course, if a voltage decay occurs and the contacts re-connect, the mechanisms are essentially the same as for the insertion transient. The key point is that no additional mechanisms have been noted for device removal that could be more disruptive than those operating during the insertion process. In the limit, the removal process could produce as much disruption as the insertion process.

Figure 22 shows two examples of demating. The demating events shown in Figure 22a have only approximately a 60-microsecond separation. This separation is exceptionally small, and it is theoretically possible to have coincidental contact-to-contact events (within the precision of the instrumentation). The demating event with bounce shown in Figure 22b was acquired on exactly the

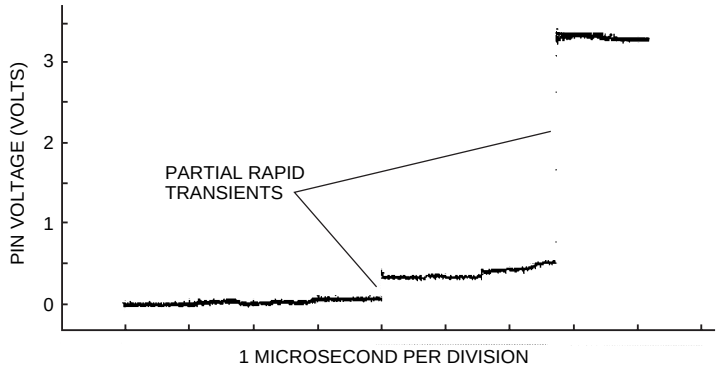


Figure 20
Gradual and Rapid Transients in the Same Mating Process

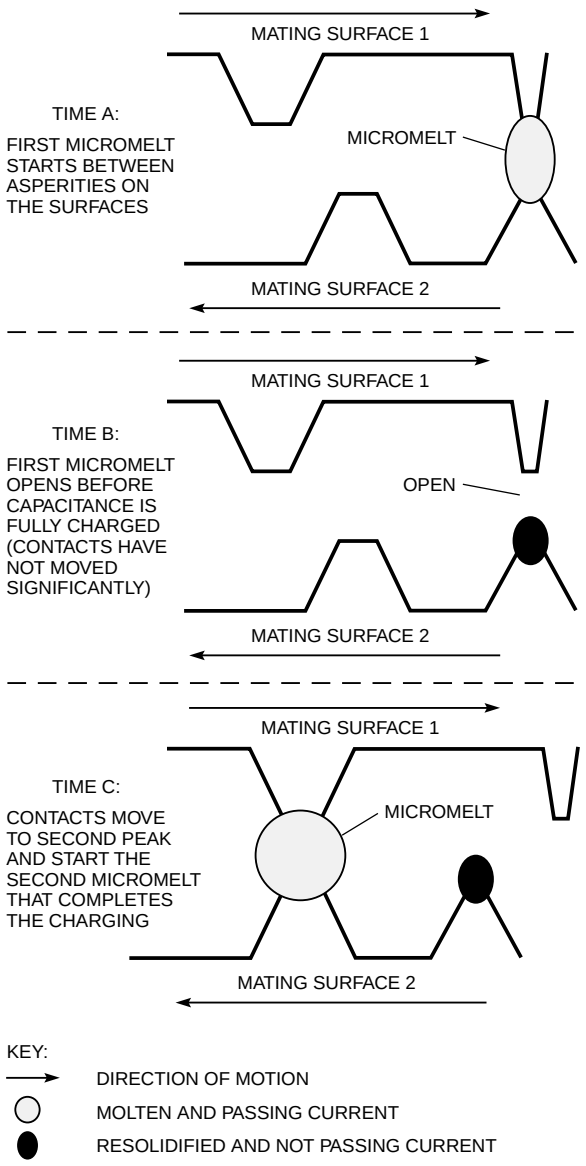


Figure 21
Architecture of Combination Gradual/Rapid Mating Event

same pins in exactly the same connector used for the events in the top of the figure, and there is no evidence of any activity on pin 2. Pin 2 demated long before any activity was seen on pin 1. Again, this underscores the unpredictability of the details of any given event.

Impact of Device Insertion and Removal on Bus Signals

This section contains several examples of the noise produced on the bus side of the connector. Actual devices with approximately 25 pF of capacitance were used to obtain the data. This capacitance value is increased by the probe capacitance. On the bus side, there is also some increased capacitance caused by the probe used to acquire the bus side signal. Figure 23 shows the basic impact of a rapid transient on the bus side of the connector and the time relationship of the bus disturbance to the voltage on the device side. The bus voltage is reduced while it supplies the necessary charge to the device pin. After the device capacitance is charged, the bus resumes its voltage level before the insertion transient (more or less).

In this test, the bus pulse is approximately 3-ns wide at its midpoint; its peak amplitude is approximately 1.25 V. This pulse is significantly larger in amplitude than that produced from a device alone.

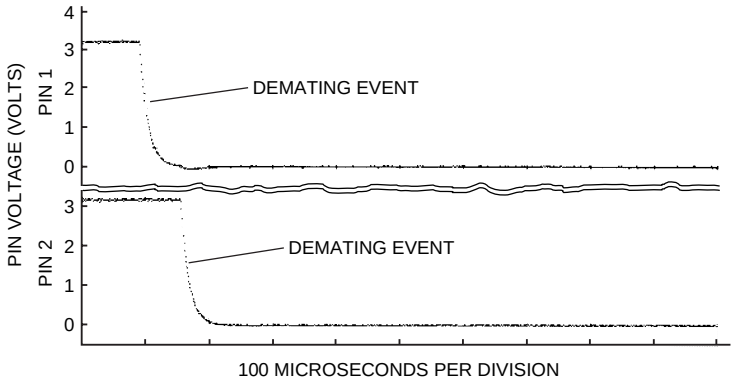
One of the more interesting features of the signals in Figure 23 is the lack of commonality or tracking in the signals after the rapid transient has passed. In the simplest interpretation, one would expect both sides of the connector to have nearly the same voltage (at the least to be within the accuracy of the 0.1-ns propagation time between the probes). The following discussion addresses the author's current thinking on the reasons for this lack of tracking.

Instrumentation effects, such as resonance or differences in probe properties, were ruled out by using both probes on the same signal and noting that there was little difference in the signals reported from each channel. Later, typically after a few microseconds, the voltages do become effectively the same.

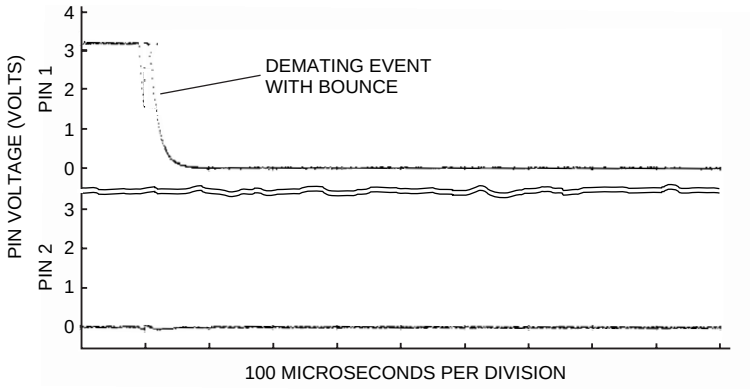
Because a significant voltage difference is present for relatively long times, there must be a significant voltage source between the contacts to support this observed difference. In the initial stages, the difference between the pin voltages is approximately 3 V. If the current is the one calculated in the section Limiting Parameters for the Rapid Transient, that is, approximately 50 mA, then the current-limiting impedance must be at least 3/0.05 = 60 ohms. This impedance, coupled with the parasitic capacitances and inductances, serves to blunt the instantaneous electrical energy transfers that would be implied by a very low source impedance. If the source impedance were very low, then both sides would have to track shortly after the initial contact.

Part of this limiting impedance is the loaded or local transmission line impedance of the bus. The characteristic impedance is nominally approximately 100 ohms for an unloaded bus. Since the bus connector is attached to the middle of the line, both sides are available to supply charge and the effective charging impedance would be approximately 50 ohms. A 30-pF capacitance would have a charging time constant of 1.5 ns. This time constant fits the observations well during the rapid transient itself but does not fit the timing parameters of the voltage differences observed well after the rapid transient.

Elevated local temperatures are almost certainly present during the rapid transient (near the melting point of the metals!), so it seems plausible that the mystery voltage source is basically thermal electromotive force (EMF) between the pins. Allowing a few microseconds to achieve thermal equilibrium and subsequent loss of the thermal EMF also seems quite plausible. These details are inviting further detailed investigation but



(a) Demating Event with 60-microsecond Separation



(b) Demating Event with Bounce

Figure 22
Demating Events

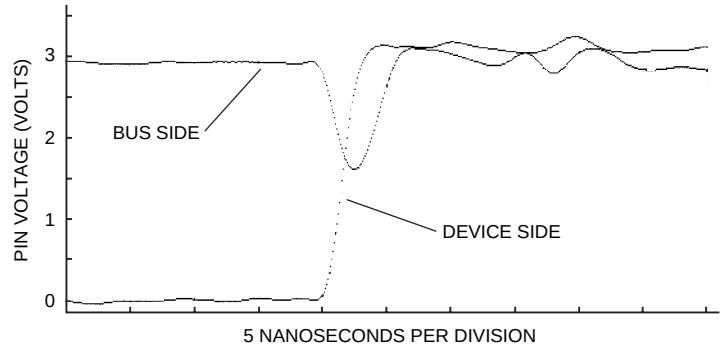


Figure 23
Most Severe Noise Pulse Observed

do not affect the practical conclusions as applied to parallel SCSI.

As added evidence for thermal effects, experiments with early LVD SCSI devices that use a 1.2-V bus level instead of the 3.5-V bus level shown in Figure 23 transfer much less energy and have a much shorter settling time before both sides of the contact track. These LVD results will be reported separately.

The point extracted from these charging-impedance and settling-time observations is simply that the overall energy transfer rate is limited by the microphysics of the process. This means that Figure 23 almost certainly illustrates the worst-case disturbances.

It has been noted that the bus pulse is similar to that produced by a stub on the bus and a signal with a fast rise/fall time. In a sense, we really are charging a stub in either case,

and in both cases the loaded or local characteristic impedance of the bus limits the extent of the disturbance.

To more accurately measure the noise pulse produced when a device is added to the bus, measurements were performed without a scope probe attached to the device pin. To do so required triggering the scope from the noise pulse on the bus side. Consequently, it was not possible to see the device-side charging dynamics. Figure 24 shows the measured pulse near the device connector and at a point 2 meters away.

The pulse measured in Figure 24 has approximately half the amplitude of the pulse in Figure 23. This is more reduction in amplitude than one would expect from the removal of 10 pF from the effective device capacitance, and this difference, while not completely explained, is in the favorable direction. The noise pulse that reached the next device (where it could be detected as an error) would be even smaller, because of the dispersion and attenuation in the bus and because the neighboring device would need to have its 25-pF capacitance charged also. The signal at the measurement point 2 meters away in Figure 24 indicates the intensity of the attenuation and dispersion to be expected in the rapid transient bus pulses. The details of the attenuation and dispersion depend somewhat on the bus media used.

The rapid transient bus pulses are shown on actual data pulses in Figure 25. The top trace in the figure shows a rapid transient pulse on a negated part of a single-ended SCSI signal. There is a scope probe on this device, but the device capacitance is only approximately 15 pF so the total with the probe is approximately 25 pF. Note that the noise pulse is approximately 0.8 V and does not take the signal into the receiver detection range below 2 V. This negated state is a bit higher than usually found, so the bus pulse is starting from a higher point. If the pulse had started from a lower point, for example about 2.5 V, the pulse amplitude would not have been as large. Further discussion of the receiver detection range appears later in this section.

The signals in Figure 25 were purposely chosen to have broad falling edges of approximately 15 ns. Normal SCSI signals are 5 ns or faster. The broad edges maximize the chance that the bus pulse will produce a signal slope reversal of the type that can produce multiple edges. The bottom trace in Figure 25 shows a bus pulse in the most sensitive part of the falling edge. This pulse produces almost no slope reversal because by the

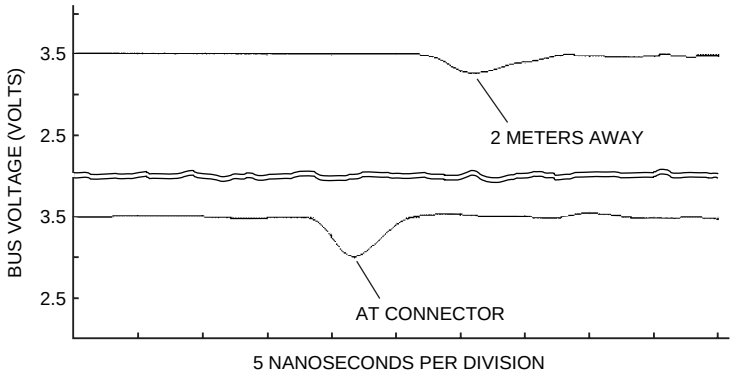


Figure 24
Bus Pulses with No Scope Probe Attached to the Device

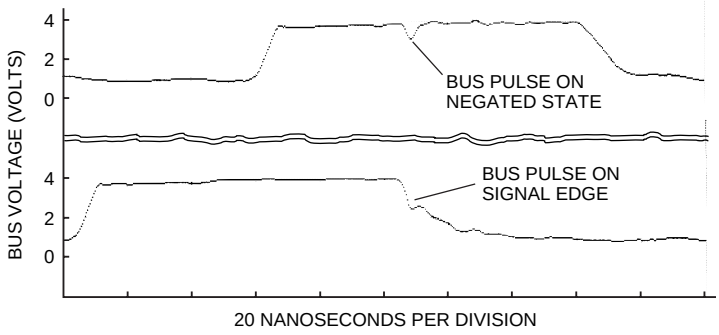


Figure 25
Bus Pulses on Actual Signals

time it is ready to become positive-going, the data signal has fallen so much that there is no voltage source to drive the signal more positive. At the beginning of the falling edge, the slew rate is increased by the bus pulse; in the middle, the edge is extended and consequently the overall time required for the falling edge is almost exactly the same as for the falling edge that has no bus pulse (see the top trace).

Therefore, the main effect of rapid transient pulses occurs when they intersect the signal edges (where a state change is expected anyway), and the effect is movement of the position of the edge by no more than 2 ns from the normal position. This movement is already accounted for in the SCSI standard as *pulse distortion skew,* so there is no important effect.

If the mating event happens while the bus signal is in the asserted state, there is little effect since little charge is transferred. If the event happens in the rising edge, there may not be enough voltage difference to start a rapid transient—again, there is little effect. If a rapid transient is initiated on a rising edge, the impact is still a small shift in the position of the edge. In any arbitrary combination of signal level and type of transient, the bus disturbance will not be greater than those shown in Figure 24 and Figure 25.

Differential

For differential SCSI systems, essentially the same behavior occurs as for the single-ended case except that the relationship between two contacts instead of just one must be considered. If insertion transients on the positive signal differential line are occurring at the same time as transients on the negative signal line, we must examine the difference between these transients to see what impact they have on the differential signals. Based on the time required between mating events on neighboring connector pins presented in the section Connector Insertion Dynamics and in Figure 15, it is evident that the differential case is almost always two independent and isolated single-ended cases. This is because the difference in the time required for different pins in the same connector to begin the mating process vastly exceeds the actual transient time on either signal.

In SCSI differential systems, both the positive and the negative signals are normally positive with respect to the local grounds. This means that the transients will be the same polarity on both signals.

In the very rare cases in which some overlap exists between the transient times on both signals, the rapid transient disturbances would usually be seen as common mode events that reduce the effective differential transient signal. These events are not seen if common mode noise exists where the signals have opposite polarity with respect to local grounds during the transients. In this case, it is theoretically possible to produce anticommon mode differential transients. However, the anticommon mode case will always have the positive and negative signal lines within a differential logical voltage level of ground, and the transients will therefore be small. Even in the anticommon mode case, the effect is at most a slight shift in the time when the differential state change is observed, since the transient disturbances are so small.

In the pathological differential case, large common mode levels exist on both the positive and the negative signals. The insertion transient will be larger because the bus voltage is larger. This case is even more rare since it requires both coincidental pin mating and coincidental large common mode.

The other case considered that can have a unique effect on differential systems is that of extended bounce. This case extends the effective mating time to the point when some overlap between the transient activity on the pins is more likely. Recall that the extended bounce case was only visible when a leakage mechanism was available to discharge the incoming device capacitance. In actual devices, no significant leakage occurs so a bounce event does not produce disturbances after this first contact.

The differential signal seen by the incoming device may be seriously affected by extended bounce if there is bus activity during this bounce. Consider, for example, a case in which the positive signal contact opened because of a bounce event after achieving a full charge. While it is still open, the negative signal changes state. Now both the positive and negative signals are at the same nominal potential, which is an indeterminate differential condition. Fortunately, this condition is not a problem because the only device that sees this condition is the device being inserted or removed and it is not in an operational state.

Summary of the Handling of Device Insertion and Removal Transients

After a complex, yet self-consistent, set of experimental data and interpretations, the concluding results are that the worst-case SCSI bus transients resulting from proper insertion and removal processes should not cause errors in the SCSI bus as presently specified in the SPI and Fast-20 (UltraSCSI) standards. The proper processes include pregrounding prior to insertion, avoiding excessive device capacitance, and using SCSI drivers and receivers that meet all of the SCSI requirements.[4]

As of this writing, all reports of device insertion/removal errors have been traced back to failure to use proper procedures or designs. The most common errors are lack of pregrounding, devices that do not maintain the high-impedance input state during power cycling, and power distribution or mechanical transient effects unrelated to SCSI proper.

The mechanisms that operate span a time spectrum from picoseconds in rapid transients to seconds in contact wipe and other macro connector operations.

The worst-case differential transients occur when one treats the differential system as two independent single-ended SCSI buses—one for the positive signal and one for the negative signal.

The rapid transient becomes more and more detectable as bus speeds increase and the receivers and timing margins become more sensitive. Schemes to encourage the gradual transient are the best protection against the ultimate problems caused by rapid transients. The best-known method for producing reliable gradual transients is to avoid a metal-to-metal contact during the initial contact and until the device capacitance is charged. At this time, no such connector system exists for SCSI applications.

Overall Summary

Evolution in four significant hardware technologies in the recent past has enabled parallel SCSI to break through the barriers that were preventing it from delivering excellent value, flexibility, and growth to the computer data storage industry. Application of more scientific methods, use of the latest silicon technology, and developments in the interconnect technology provided the foundation for these improvements. DIGITAL provided most of the basic data and led important standards and industry bodies to accomplish this.

Acknowledgments

Fee Lee, Keith Childs, Chuck Bagg, Pak Seto, and Jonathan Salles were instrumental in developing the special test environment and for acquiring much of the data presented in this paper. The author gratefully acknowledges the management and technical support from Pete Korce, Mike Chamberlain, Bob Passmore, Richie Lary, Ken Chester, Bob Rennick, Laura Woodburn, and Ellen Lary.

References

1. *ANSI X3.277-1996: Information Technology X3T9.2/375R SCSI-3 Fast-20, X3T10/1071D* (New York: American National Standards Institute, 1996). Available from Global Engineering, 15 Inverness Way East, Englewood, CO 80112-5704, tel. (800) 854-7159 or (303) 792-2181, fax (303) 792-2192.
2. Very High Density Cabled Interconnect (VHDCI) Specification, SFF-8441. Available from the SFF Committee, 14426 Black Walnut Court, Saratoga, CA 95070, voice fax-back service (408) 741-1600, tel. (408) 867-6630 ext. 303, fax (408) 867-2115.
3. *ANSI X.3.131-1994: Information Systems—Small Computer System Interface–2* (*SCSI-2*), X3T9.2/375R (New York: American National Standards Institute, 1994). Available from Global Engineering, 15 Inverness Way East, Englewood, CO 80112-5704, tel. (800) 854-7159 or (303) 792-2181, fax (303) 792-2192.
4. *ANSI X3.253-1995: Information Technology—SCSI-3 Parallel Interface* (*SPI*), *X3T10/855D* (New York: American National Standards Institute, 1995). Available from Global Engineering, 15 Inverness Way East, Englewood, CO 80112-5704, tel. (800) 854-7159 or (303) 792-2181, fax (303) 792-2192.
5. *SCSI Enhanced Parallel Interface* (*EPI*), *X3T10/1143D* (New York: American National Standards Institute, 1997). Available from Global Engineering, 15 Inverness Way East, Englewood, CO 80112-5704, tel. (800) 854-7159 or (303) 792-2181, fax (303) 792-2192.
6. Information about I/O interfaces is available at the T10 home page at http://www.symbios.com/t10.
7. Single Connector Attachment–2 (SCA-2) Specifications: SFF-8015, SFF-8046, SSF-8048, SSF-8066, and SSF-8451. Available from the SFF Committee, 14426 Black Walnut Court, Saratoga, CA 95070, voice fax-back service (408) 741-1600, tel. (408) 867-6630 ext. 303, fax (408) 867-2115.

Biography



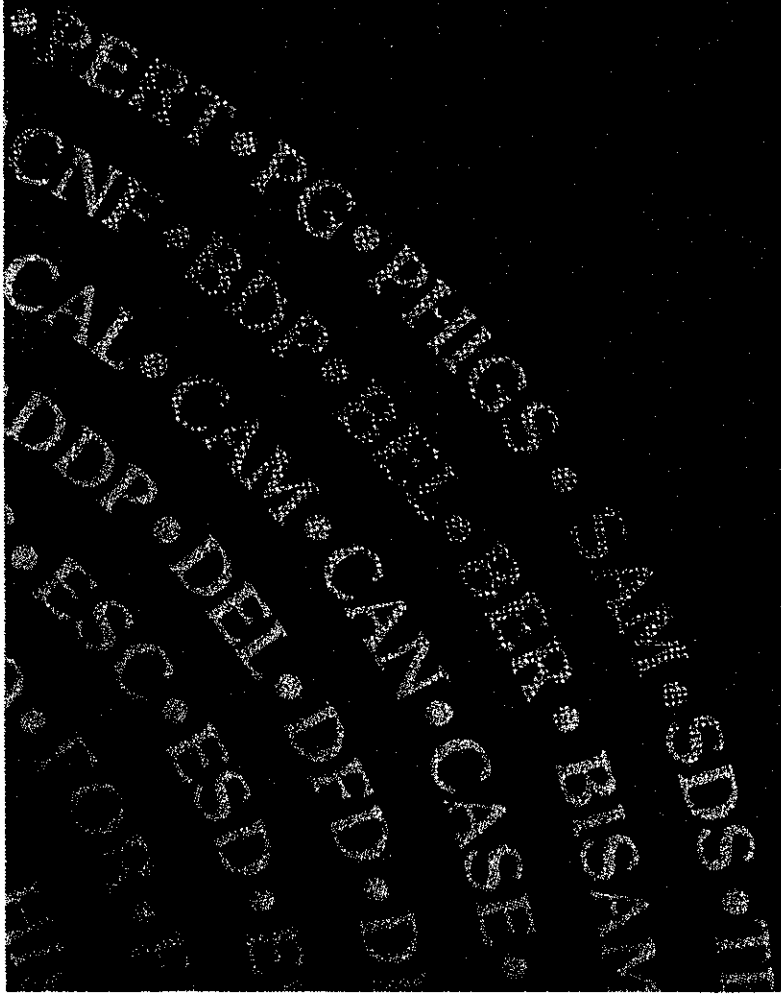
William E. Ham

Bill Ham joined DIGITAL in 1983 and has been working in storage technology since 1990 as the manager of the Storage Bus Technical Office (SBTO). The SBTO group represents DIGITAL at most of the important standards and industry bodies that are involved with the transmission of storage data. Presently, these groups are SCSI, STA, Fibre Channel, and SFF. The SBTO group also creates information from actual laboratory testing and was responsible for introducing Fast-20 technology to ANSI in 1993 and hot-plugging technology to SCSI in 1992. Bill has a Ph.D. in electrical engineering from Southern Methodist University in Dallas, Texas, and has been active in the electronics industry in a variety of technologies since 1970. He has held significant positions in silicon chip, printed circuit board, multichip module technology, and storage bus technology. In addition, he is the past technical editor of the SSA physical standards, past editor of the new SPI-2 SCSI standard, editor of the entire family of SFF connector documents, editor of the new Enhanced Parallel Interface ANSI project for SCSI, and secretary for the Fibre Channel physical standards group (T11.2).

Exhibit G

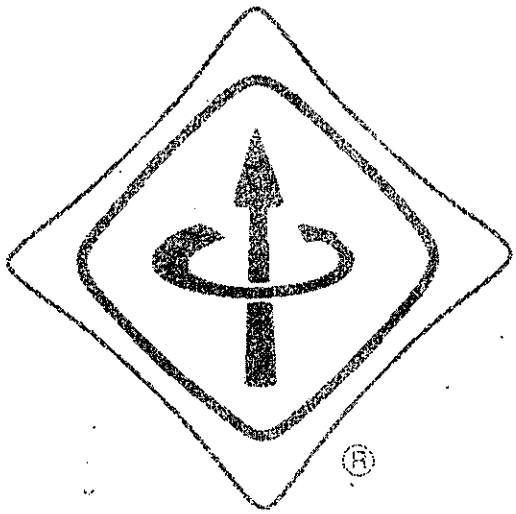
The New IEEE Standard Dictionary of Electrical and Electronics Terms

Fifth Edition
Newly Revised and Expanded



IEEE

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



IEEE

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

Copyright © 1993 by the
Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 1993
Printed in the United States of America

ISBN 1-55937-240-0

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

January 15, 1993

SH15594

drift velocity will have a direction opposite to the electric field. 457-1982

drift, zero. Drift with zero input. *See:* **electronic analog computer.** 165-1977

drill and practice interaction. An instruction method employed by some computer-assisted instruction systems, in which the student is asked repeatedly to perform the same or similar tasks. 610.2-1987

dripproof. So constructed or protected that successful operation is not interfered with when faling drops of liquid or solid particles strike or enter the enclosure at any angle from 0 to 15 degrees from the downward vertical unless otherwise specified. *See:* **asynchronous machine.** 45-1983, C37.100-1981, [9]

dripproof enclosure (1) (metal-enclosed bus and calculating losses in isolated-phase bus). An enclosure usually for indoor application, so constructed or protected that falling drops of liquid or solid particles that strike the enclosure at any angle not greater than 15 degrees from the vertical will not be able to interfere with the successful operation of the metal-enclosed bus. C37.23-1987
(2) (electric installations on shipboard). An enclosure in which the openings are so constructed that drops of liquid or solid particles falling on the enclosure at any angle not greater than 15 degrees from the vertical either cannot enter the enclosure, or if they do enter the enclosure, they will not prevent the successful operation of, or cause damage to, the enclosed equipment. 45-1983
(3) (power and distribution transformer). An enclosure, usually for indoor application, so constructed or protected that falling drops of liquid or solid particles that strike the enclosure at any angle within a specified variation from the vertical shall not interfere with the successful operation of the enclosed equipment. C57.12.80-1978

dripproof machine. An open machine in which the ventilating openings are so constructed that drops of liquid or solid particles falling on the machine at any angle not greater than 15 degrees from the vertical cannot enter the machine either directly or by striking and running along a horizontal or inwardly inclined surface. *See:* **asynchronous machine.** [119]

driptight (transformer). So constructed or protected as to exclude falling dirt or drops of liquid, under specified test conditions. C57.12.80-1978

driptight enclosure. An enclosure so constructed that falling drops of liquid or solid particles striking the enclosure at any angle within a specified variation from the vertical cannot enter the enclosure either directly or by striking and running along a horizontal or inwardly inclined surface. C37.30-1971, C37.100-1981, C57.12.80-1978

drive (1) (industrial control). The equipment used for converting available power into mechanical power suitable for the operation of a machine. *See:* **electric drive.** 317-1976, [75]
(2) (electronic computation and recording). *See:* **tape drive.**
(3) (NuBus®). A module activity causing a bus signal line to be in a particular sate. 1196-1987

® NuBus is a registered trademark of Texas Instruments, Inc.

driven element (antenna). A radiating element coupled directly to the feed line. *See:* **antenna.** [35]

drive pattern (facsimile). Density variation caused by periodic errors in the position of the recording spot. When caused by gears this is called gear pattern. *See:* **recording (facsimile).** 167-1966

drive pin (disk recording). A pin similar to the center pin, but located to one side thereof, that is used to prevent a disc record from slipping on the turntable. *See:* **phonograph pickup.** [32]

drive-pin hole (disk recording). A hole in a disc record that accommodates the turntable drive pin. [32]

drive pulse (static magnetic storage). A pulsed magnetomotive force applied to a magnetic cell from one or more sources. 163-1959w

driver (1) (communication practice). An electronic circuit that supplies input to another electronic circuit. [119]
(2) (software). (A) A software module that invokes and, perhaps, controls and monitors the execution of one or more other software modules. *See also:* **test driver.** (B) A computer program that controls a peripheral device and, sometimes, reformats data for transfer to and from the device. 610.12-1990

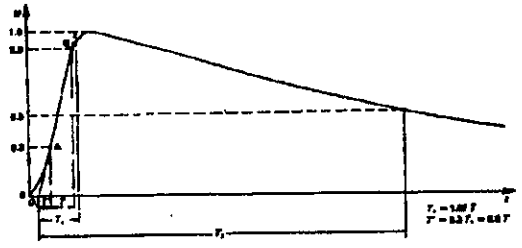
drive strip (chafing strip) (rotating machinery). An insulating strip located in the coil slots between the wedge and the top of the slot armor or the top coil side, to provide protection during assembly of the wedges. *See:* **stator.** [9]

driving agent. The agent that is permitted to assert or negate a signal on the bus. 1296-1987

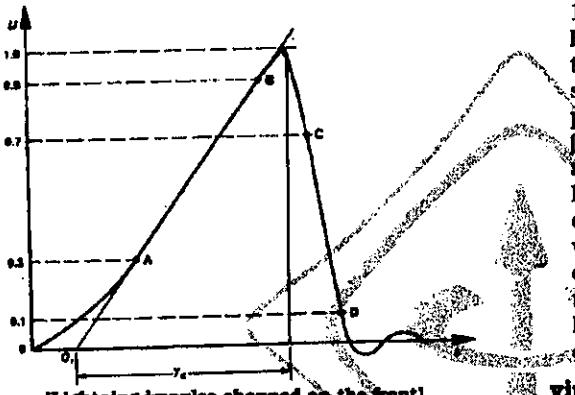
driving edge. A time corresponding to a rising edge of the bus clock. 1196-1987

driving machine. The power unit that applies the energy necessary to raise and lower an elevator or dumbwaiter car or to drive an escalator or a private-residence inclined lift. [119]

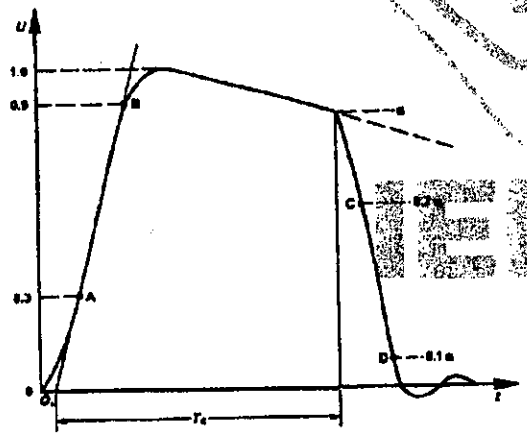
driving-point admittance (between the *j*th terminal and the reference terminal of an *n*-terminal network). The quotient of (1) the complex alternating component I_j of the current flowing to the *j*th terminal from its external termination by (2) the complex alternating



[Full lightning impulse]



[Lightning impulse chopped on the front]



[Lightning impulse chopped on the tail]

Virtual Front Time T_1 (2)

virtual instant of chopping (voltage testing). The instant preceding point C on the figures (under virtual front time), by 0.3 times the (estimated) virtual time of voltage collapse during chopping. *See:* **test voltage and current.** [55]

virtual junction temperature. The temperature of the active semiconductor element of a semiconductor device based on a simplified representation of the thermal and electrical behavior of the device. It is particularly applicable to multi-junction semiconductor devices. [12]

virtual machine (software). A functional simulation of a computer and its associated devices. *See:* **computer; simulation.** 729-1983

virtual memory. *See:* **virtual storage.** 610.12-1990

virtual origin (impulse current or voltage). *See:* **virtual zero time.**

virtual origin O_1 (1) (high voltage testing). The virtual origin O_1 of an impulse current is the instant preceding that at which the current is 10 percent of the peak value by a time $0.1 \times T_1$. For oscillograms having linear time sweeps, this is the intersection with the *X* axis of a straight line drawn through the 10 and 90 percent reference points on the front. 4-1978

(2) (lightning impulse tests, general applicability). The virtual origin O_1 of a lightning impulse is the instant preceding that corresponding to point A (see figures under virtual front time) by a time $0.3 \times T_1$. For oscillograms having linear time sweeps, this is the intersection with the *X* axis of a straight line drawn through reference points A and B on the front. 4-1978

virtual peak value. *See:* **peak value.**

virtual point picture character. *See:* **radix point character.** 610.5-1990

virtual rate of rise of the front (impulse voltage). The quotient of the peak value and the virtual front time. *Note:* The term peak value is to be understood as including the term virtual peak value unless otherwise stated. *See:* **test voltage and current.** [55]

virtual record. A record that appears to be but is not physically stored; rather, it is constructed or derived from existing data when its contents are requested by an application program. 610.5-1990

virtual relation. A relation that is not stored in a database in the form in which the user sees it, but is instead derived from base relations using user-defined operations. 610.5-1990

virtual resource (ATLAS). A notational test resource, the performance characteristics of which conform to a summary of related test requirements. 771-1989

virtual sequential access method (VSAM). An access method for direct or sequential access to data records on storage devices in which auxiliary storage can be addressed as though it were part of main storage. Pages of data are transferred as needed between auxiliary and main storage. *See also:* **basic sequential access method; indexed sequential access method; queued sequential access method.** 610.5-1990

Exhibit H

IEEE 100

THE AUTHORITATIVE DICTIONARY

IEEE
Published by
Standards Information Network
IEEE Press

R. VADEN
RECEIVED
JUN 21 2006
IRC HOGAN & HARTSON

IEEE 100
The Authoritative Dictionary of IEEE Standards Terms

Seventh Edition

IEEE
Published by
Standards Information Network
IEEE Press

IEEE believes the information in this publication is accurate as of its publication date; such information is subject to change without notice. IEEE is not responsible for any inadvertent errors.

Other tradenames and trademarks in this document are those of their respective owners.

The Institute of Electrical and Electronics Engineering, Inc.
3 Park Avenue, New York, NY, 10016-5997, USA

Copyright © 2000 by the Institute of Electrical and Electronics Engineers, Inc. All rights reserved. Published December 2000. Printed in the United States of America.

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

To order IEEE Press publications, call 1-800-678-IEEE.

Print: ISBN 0-7381-2601-2 *SP1122*

See other standards and standards-related product listings at: http://standards.ieee.org/

The publisher believes that the information and guidance given in this work serve as an enhancement to users, all parties must rely upon their own skill and judgement when making use of it. The publisher does not assume any liability to anyone for any loss or damage caused by any error or omission in the work, whether such error or omission is the result of negligence or any other cause. Any and all such liability is disclaimed.

This work is published with the understanding that the IEEE is supplying information through this publication, not attempting to render engineering or other professional services. If such services are required, the assistance of an appropriate professional should be sought. The IEEE is not responsible for the statements and opinions advanced in this publication.

Library of Congress Cataloging-in-Publication Data

IEEE 100 : the authoritative dictionary of IEEE standards terms.—7th ed.
p. cm.
ISBN 0-7381-2601-2 (paperback : alk. paper)
1. Electric engineering—Dictionaries. 2. Electronics—Dictionaries. 3. Computer engineering—Dictionaries. 4. Electric engineering—Acronyms. 5. Electronics—Acronyms. 6. Computer engineering—Acronyms. I. Institute of Electrical and Electronics Engineers.

TK9 .I28 2000
621.3'03—dc21 00-050601

Case 1:07-mc-00493-RMC Document 188-9 Filed 07/23/2008 Page 5 of 5

(14) A reference to an integrated circuit or other design structure. (C/TT) 1450-1999

device address The (32-m)-bit identifying number assigned to a FASTBUS device that is compared with the signals on the AD lines during a logical primary address cycle of a FASTBUS operation. The device address is formed by the group and module address fields. The (remaining) low-order m bits are assigned to the internal address field. (NID) 960-1993

device alias A shorthand representation for a device path. (C/BA) 1275-1994

device arguments The component of a node name that is provided to a package's open method to provide addtional device-specific information. (C/BA) 1275-1994

device class-broadcast Selective broadcast-class specified by CSR#7. Controls device response to subsequent cycles within the broadcast. (NID) 960-1993

device communications controller (DCC) A communications interface associated with a medical device. A DCC may support one or more physically distinct devices acting as a single network communications unit. Its purpose is to provide a point-to-point serial communication link to a bedside communications controller (BCC). (EMB/MIB) 1073.4.1-2000, 1073.3.2-2000

device control character (data management) A control character used for the control of auxiliary devices associated with a data processing system or data communication system; for example, a control character for switching such devices on or off. (C) 610.5-1990w

device control language A language used to monitor and/or control the state of a device. (C/MM) 1284.4-2000

device coordinate system (computer graphics) A device-dependent coordinate system in which the coordinates of addressable points are expressed in integer addressable units. *Note:* A device driver maps normalized device coordinates or world coordinates to actual device coordinates. (C) 610.6-1991w

device-dependent (computer graphics) Pertaining to that which can be used only on a particular device. *Contrast:* device-independent. (C) 610.6-1991w

device driver (1) (computer graphics) The software that translates device-independent commands into device-specific commands. (C) 610.6-1991w

(2) The software responsible for managing low-level I/O operations for a particular hardware device or set of devices. Contains all the device-specific code necessary to communicate with a device and provides a standard interface to the rest of the system. *See also:* firmware device driver; operating system device driver. (C/BA) 1275-1994

(3) A program that runs on the host and manages the sending and receiving of information from the peripheral. The driver utilizes the link level interface defined in this standard to communicate data between the application program and the peripheral personality. (C/MM) 1284-1994

(4) A software component that permits a system to control and communicate with a peripheral device. *See also:* printer driver; disk driver. (C) 610.10-1994w

Device ID A structured, variable length ASCII message identifying the manufacturer, command set, and model of the peripheral. The message is provided by the peripheral in response to a request issued by the host during the negotiation phase. Provided that the peripheral supports the bidirectional mode requested by the host, this message is provided in the requested mode. The Device ID is intended to assist the host in selecting the device and/or peripheral driver appropriate to the peripheral. (C/MM) 1284-1994

device-independent (computer graphics) Pertaining to that which can be used on a variety of devices. *Contrast:* device-dependent. (C) 610.6-1991w

device interface One of the interfaces specified in this standard that allows devices to be identified, characterized, and used to assist other Open Firmware functions such as booting. (C/BA) 1275-1994

device media control language (data management) A language that may be used to describe the physical layout and organization of data within some physical storage media. (C) 610.5-1990w

device node A particular entry in the device tree, usually describing a single device or bus, consisting of properties, methods, and private data. (A device node may have multiple child nodes and has exactly one parent node. The root node has no parent node.). (C/BA) 1275-1994

device path A textual name identifying a device node by showing its position in the device tree. (C/BA) 1275-1994

device register (A) An addressable register used to store information describing the device. *See also:* control register. **(B)** An addressable register used to store status and control information, and data for transmission to or from a device. *Synonym:* device status word. (C) 610.10-1994

device rise time (photomultipliers for scintillation counting) The mean time difference between the 10- and 90-percent amplitude points on the output waveform for full cathode illumination and delta-function excitation. DRT is measured with a repetitive delta-function light source and a sampling oscilloscope. The trigger signal for the oscilloscope may be derived from the device output pulse, so that light sources such as the the scintillator light source may be employed. (NPS) 398-1972r

device space (computer graphics) The area defined by the addressable points of a display device. (C) 610.6-1991w

device specifier Either a device path, a device alias, or a hybrid path that begins with a device alias and ends with a device path. (C/BA) 1275-1994

device status word *See:* device register.

device tree A hierarchical data structure representing the physical configuration of the system. (The device tree describes the properties of the system's devices and the devices' relationships to one another. Most Open Firmware elements [devices, buses, libraries of software procedures, etc.] are named and located by the device tree.). (C/BA) 1275-1994

dew point The temperature at which the water vapor in the gas begins to condense, expressed in degrees Fahrenheit (°F) or Celsius (°C). (PE/IC) 1125-1993

device port The physical connection points through which signals flow into or out of a device or where timing, synchronization, and triggering control are accomplished. (SCC20) 993-1997

device type Identifies the set of properties and package classes that a node is expected to implement. Specified by the "device_type" property. (C/BA) 1275-1994

device under test (DUT) The device to be placed in a test fixture and tested. (C/TT) 1450-1999

dew point temperature The temperature at which condensation of water vapor begins in a space. (IA/PSE) 241-1990r

dew withstand voltage test A test to determine the ability of the insulating system to withstand specified overvoltages for a specified time without flashover or puncture while completely covered with dew. (SWG/PE) C37.100-1992, C37.23-1987r

dezincification Parting of zinc from an alloy (parting is the preferred term). *Note:* Other terms in this category, such as denickelification, dealuminification, demolybdenization, etcetera, should be replaced by the term parting. *See also:* parting. (IA) [59]

DF *See:* direction finder.

DF antenna *See:* direction finder antenna system.

DFD *See:* data flow diagram.

D Filter A 300 Hz to 3400 Hz bandpass filter used for measuring noise, impulse noise, or data modem signal power. Noise measured through the D-Notched filter is used to evaluate its effect on the performance of a data modem. (COM/TA) 743-1995

D flip-flop A flip-flop that has one data input, one trigger, and an output which assumes the state of the data input when the trigger is received. (C) 610.10-1994w

DFS *See:* depth-first searc

DF sensitivity *See:* direct

dg *See:* decilog.

diad (mathematics of co lated items or digits.

diagnosis (1) The conclu servations, or other inf (AT

(2) A cognitive assessn

diagnosis, fault *See:* faul

diagnostic (1) (software) lation of faults or failur a diagnostic manual.

(2) A process by whic tected.

diagnostic controller Th reasoner system or fro cedures in the sequenc

diagnostic factor (therr ment and electrical in pability) A variable c periodically or contin measure the degree of aging process.

diagnostic field tests an Procedures that are pe ratus or parts thereof i service. *Note:* The para to apparatus and may cal, thermal, etc., qua usually based on a char or by comparison with normally carried out a perience and/or manuf may also be performe determine the location

diagnostic knowledge P port the diagnostic pr lationships between po may cause these outco

Diagnostic Machine Ai used for functional te description in terms of operators, such as logi

diagnostic manual (sof information necessary system or component those malfunctions. T features of the system available for its supp port manual; user m manual.

diagnostic procedure A observations, and othe faults.

diagnostic process A str servations, and other faults.

diagnostic resolution T mum number of repla

diagnostic routine (1) malfunction in the co programmed check.

(2) **(test, measureme** ical sequence of tests unit under test.

Exhibit I

OIPE IAP14 MAR 28 2006 PATENT & TRADEMARK OFFICE

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of: Michael Tasler

Group No.: 2181

Serial No.: 11/078,778

Conf. No.: 8978

Filed: 3/11/06

Examiner: Harold J. Kim

For: ANALOG DATA GENERATING AND PROCESSING DEVICE FOR USE WITH A PERSONAL COMPUTER (As Amended)

I hereby certify that this paper is being deposited with the United States Postal Service as EXPRESS MAIL POST OFFICE TO ADDRESSEE service under 37 C.F.R. 1.10 on the date indicated below and is addressed to: Commissioner for Patents, Mail Stop – Fee Amendment, P.O. Box 1450, Alexandria, VA 22313-1450:

Date: March 28, 2006
Exp. Mail No. EV555557389US

Attorney Docket No.: 0757/96910

PRELIMINARY AMENDMENT

Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-0001

Dear Sir:

Please enter this preliminary amendment prior to examination of the above-captioned application.

03/29/2006 FMETEKI1 00000099 230920 - 11078778
01 FC:2202 - 1400.00 DA

IN THE TITLE:

Please amend the title to read as follows: ~~Flexible Interface~~Analog Data Generating And Processing Device For Use With A Personal Computer.

IN THE CLAIMS:

Please cancel claims 1-16 without prejudice as to the subject matter claimed therein, and please add the following claims 17-93:

17. (new) An analog data generating and processing device for use with a personal computer having at least one multi-purpose interface to which the personal computer sends periodic inquiry signals as to what type of device is operatively connected thereto, the analog data generating and processing device comprising:

a sensor that is mounted on a housing, the sensor being adapted to receive analog wave signals that are generated by a source that is external to the housing and that is not located in substantial proximity to the sensor, the sensor being further adapted to generate sets of analog data from the analog wave signals that it receives;

an analog to digital converter that is operatively connected to the sensor and that generates a set of digitized analog data from each set of analog data;

a circuit that includes a processor and a memory that are operatively connected to the analog to digital converter, a first set of instructions being stored in the memory that are utilized by the processor to cause the sets of digitized analog data to be individually stored in the memory irrespective of whether or not the analog data generating and processing device has been recognized by the personal computer;

an input/output port that is adapted to be operatively connected to the multi-purpose interface of the personal computer, a response signal being automatically and without user intervention sent from the input/output port to the multi-purpose interface after they have been operatively connected together and after an inquiry signal has been received by the

input/output port, the receipt and processing of the receipt signal by the personal computer causing it to automatically and without user intervention recognize the analog data generating and processing device as being a device having digital data that is stored therein and selectively retrievable therefrom; and

wherein, after the analog data generating and processing device has been automatically recognized by the personal computer, and while the input/output port is operatively connected to the multi-purpose interface, user selected ones of the digitized sets of analog data can be transferred from the memory, through the input/output port, through the multi-purpose interface, and to the personal computer by means of a driver that is associated with the personal computer.

18. (new) A combination comprising the analog data generating and processing device of claim 17 and a personal computer.

19. (new) The analog data generating and processing device of claim 17, wherein the analog wave signals comprise electromagnetic radiation.

20. (new) The analog data generating and processing device of claim 19, wherein the electromagnetic radiation received by the sensor is representative of an object that is physically separated from and can be located not in substantial proximity to the housing.

21. (new) The analog data generating and processing device of claim 20, wherein the electromagnetic radiation is generated by a medical device.

22. (new) The analog data generating and processing device of claim 21, wherein the medical device comprises a diagnostic radiological system.

23. (new) The analog data generating and processing device of claim 20, wherein the sensor comprises an electronic measuring device.

24. (new) The analog data generating and processing device of claim 23, wherein the electronic measuring device comprises a multi-meter.

25. (new) The analog data generating and processing device of claim 20, wherein the driver is adapted for use with a mass storage device.

26. (new) The analog data generating and processing device of claim 25, wherein the driver is adapted for use with a mass storage device that includes a rotatable storage medium.

27. (new) The analog data generating and processing device of claim 26, wherein the driver is adapted for use with a hard disk drive.

28. (new) The analog data generating and processing device of claim 20, wherein the driver is located in a memory of the personal computer.

29. (new) The analog data generating and processing device of claim 28, wherein the personal computer memory comprises a BIOS of the personal computer.

30. (new) The analog data generating and processing device of claim 20, wherein receipt and processing of the response signal by the personal computer allows it to communicate with the analog data generating and processing device as if it were a mass storage device even though it is not a mass storage device.

31. (new) The analog data generating and processing device of claim 30, wherein receipt and processing of the response signal by the personal computer allows it to communicate with the analog data generating and processing device as if it were a mass storage device having

a rotatable storage disk even though it is not a mass storage device having a rotatable storage disk.

32. (new) The analog data generating and processing device of claim 31, wherein receipt and processing of the response signal by the personal computer allows it to communicate with the analog data generating and processing device as if it were a hard disk drive even though it is not a hard disk drive.

33. (new) The analog data generating and processing device of claim 20, wherein the memory of the analog data generating and processing device comprises a buffer memory.

34. (new) The analog data generating and processing device of claim 20, wherein the input/output port is adapted to be operatively connected to a SCSI interface of the personal computer.

35. (new) The analog data generating and processing device of claim 20, wherein the processor comprises a digital signal processor.

36. (new) The analog data generating and processing device of claim 20, wherein the sets of digitized analog data are transferred to the personal computer in a format suitable for a mass storage device present in the personal computer.

37. (new) The analog data generating and processing device of claim 36, wherein a root directory and virtual files are created in the memory which can be accessed by the personal computer.

38. (new) The analog data generating and processing device of claim 37, wherein at least one of the virtual files comprises a configuration file stored in the memory.

39. (new) The analog data generating and processing device of claim 38, wherein the configuration file allows a user to configure the analog data generating and processing device as being a specific mass storage device.

40. (new) The analog data generating and processing device of claim 39, wherein the configuration file allows a user to configure the analog data generating and processing device as being a specific hard disk drive.

41. (new) The analog data generating and processing device of claim 20, wherein a wire based connection is used to operatively connect the input/output port to the multi-purpose interface of the personal computer.

42. (new) The analog data generating and processing device of claim 20, wherein a second set of instructions are stored in the memory which are adapted to cause the response signals to be generated.

43. (new) The analog data generating and processing device of claim 20, wherein a third set of instructions are stored in the memory that allow user selected ones of the digitized sets of analog data to be transferred to a memory of the personal computer.

44. (new) An analog data generating and processing device for use with a personal computer having at least one multi-purpose interface to which the personal computer sends periodic inquiry signals as to what type of device is operatively connected thereto, the analog data generating and processing device comprising:

means for receiving analog wave signals that are generated by a source external to and not located in substantial proximity to the analog data generating and processing device, for generating sets of analog data therefrom, and for digitizing each set of analog data;

means for causing the digitized sets of analog data to be individually stored in a memory irrespective of whether or not the analog data generating and processing device has been recognized by the personal computer;

means for receiving from the multi-purpose interface of the personal computer the periodic inquiry signals, and for automatically and without user intervention responding thereto by sending a signal to the multi-purpose interface that causes the personal computer to automatically and without user intervention recognize the analog data generating and processing device as being a device having digital data that is stored in and to be selectively retrievable from a memory in which digital signals are stored; and

means for transferring user selected ones of the digitized sets of analog data to the personal computer by means of a driver that is associated with the personal computer.

45. (new) A combination comprising the analog data generating and processing device of claim 44 and a personal computer.

46. (new) The analog data generating and processing device of claim 44, wherein the analog wave signals comprise electromagnetic radiation.

47. (new) The analog data generating and processing device of claim 46, wherein the electromagnetic radiation is representative of an object that is physically separated from and can be located not in substantial proximity to the analog data generating and processing device.

48. (new) The analog data generating and processing device of claim 47, wherein the means for receiving analog wave signals forms a part of a medical device.

49. (new) The analog data generating and processing device of claim 48, wherein the medical device comprises a diagnostic radiological system.

50. (new) The analog data generating and processing device of claim 47, wherein the means for receiving analog wave signals includes an electronic measuring device.

51. (new) The analog data generating and processing device of claim 50, wherein the electronic measuring device comprises a multi-meter.

52. (new) The analog data generating and processing device of claim 47, wherein the driver is adapted for use with a mass storage device.

53. (new) The analog data generating and processing device of claim 52, wherein the driver is adapted for use with a mass storage device that includes a rotatable storage medium.

54. (new) The analog data generating and processing device of claim 53, wherein the driver is adapted for use with a hard disk drive.

55. (new) The analog data generating and processing device of claim 47, wherein the driver is located in a memory of the personal computer.

56. (new) The analog data generating and processing device of claim 55, wherein the personal computer memory comprises a BIOS of the personal computer.

57. (new) The analog data generating and processing device of claim 47, wherein receipt and processing of the response signal by the personal computer allows it to communicate with the analog data generating and processing device as if it were a mass storage device even though it is not a mass storage device.

58. (new) The analog data generating and processing device of claim 57, wherein receipt and processing of the response signal by the personal computer allows it to communicate with the analog data generating and processing device as if it were a mass storage device having

a rotatable storage disk even though it is not a mass storage device having a rotatable storage disk.

59. (new) The analog data generating and processing device of claim 58, wherein receipt and processing of the response signal by the personal computer allows it to communicate with the analog data generating and processing device as if it were a hard disk drive even though it is not a hard disk drive.

60. (new) The analog data generating and processing device of claim 47, wherein the memory of the analog data generating and processing device comprises a buffer memory.

61. (new) The analog data generating and processing device of claim 47, wherein the means for receiving from the multi-purpose interface is adapted to be operatively connected to a SCSI interface of the personal computer.

62. (new) The analog data generating and processing device of claim 61, wherein the means for transferring comprises at least a portion of a digital signal processor.

63. (new) The analog data generating and processing device of claim 47, wherein the sets of digitized analog data are transferred to the personal computer in a format suitable for a mass storage device present in the personal computer.

64. (new) The analog data generating and processing device of claim 63 wherein a root directory and virtual files are created in the memory which can be accessed by the personal computer.

65. (new) The analog data generating and processing device of claim 64, wherein at least one of the virtual files comprises a configuration file stored in the memory.

66. (new) The analog data generating and processing device of claim 64, wherein the configuration file allows a user to configure the analog data generating and processing device as being a specific mass storage device.

67. (new) The analog data generating and processing device of claim 65, wherein the configuration file allows a user to configure the analog data generating and processing device as being a specific hard disk drive.

68. (new) The analog data generating and processing device of claim 47, wherein a wire based connection is used to operatively connect the multi-purpose interface of the personal computer with the means for receiving from the multi-purpose interface.

69. (new) An analog data generating and processing device for use with a personal computer having at least one multi-purpose interface to which the personal computer sends periodic inquiry signals as to what type of device is operatively connected thereto, the analog data generating and processing device comprising:

a circuit that includes a sensor and an analog to digital converter, the circuit being adapted to be exposed to analog wave signals originate from a source that is external to the analog data generating and processing device and that is not located in substantial proximity to the sensor, to generate sets of analog data therefrom, and to generate digitized sets of analog data from the sets of analog data;

a processor and a memory both of which are operatively connected to the circuit, the processor being adapted to cause the digitized sets of analog data to be individually stored in the memory irrespective of whether or not the analog data generating and processing device has been recognized by the personal computer;

a connecting device operatively connected to the processor and the memory, the connecting device being adapted to be operatively connected to the multi-purpose interface of the personal computer and to receive therefrom the periodic inquiry signals;

wherein a response signal is automatically and without user intervention sent to the multi-purpose interface of the personal computer after the connecting device is operatively connected to the multi-purpose interface and after the connecting device receives at least one inquiry signal therefrom, receipt and processing of the response signal by the personal computer causing the personal computer to automatically and without user intervention recognize the analog data generating and processing device as being a device having digital data that is stored therein and selectively retrievable therefrom; and

wherein, after the analog data generating and processing device has been automatically recognized by the personal computer, and when the processor and memory are operatively connected to the circuit, user selected ones of the digitized sets of analog data can be transferred to the personal computer by means of a driver that is associated with the personal computer.

70. (new) A combination comprising the analog data generating and processing device of claim 69 and a personal computer.

71. (new) The analog data generating and processing device of claim 69, wherein the analog wave signals comprise electromagnetic radiation.

72. (new) The analog data generating and processing device of claim 71, wherein the electromagnetic radiation is representative of an object that is physically separated from and can be located not in substantial proximity to the analog data generating and processing device.

73. (new) The analog data generating and processing device of claim 72, wherein the electromagnetic radiation is generated by a medical device.

74. (new) The analog data generating and processing device of claim 73, wherein the medical device comprises a diagnostic radiological system.

75. (new) The analog data generating and processing device of claim 72, wherein the sensor comprises an electronic measuring device.

76. (new) The analog data generating and processing device of claim 75, wherein the electronic measuring device comprises a multi-meter.

77. (new) The analog data generating and processing device of claim 72, wherein the driver is adapted for use with a mass storage device.

78. (new) The analog data generating and processing device of claim 77, wherein the driver is adapted for use with a mass storage device that includes a rotatable storage medium.

79. (new) The analog data generating and processing device of claim 78, wherein the driver is adapted for use with a hard disk drive.

80. (new) The analog data generating and processing device of claim 72, wherein the driver is located in a memory of the personal computer.

81. (new) The analog data generating and processing device of claim 80, wherein the personal computer memory comprises a BIOS of the personal computer.

82. (new) The analog data generating and processing device of claim 72, wherein receipt and processing of the response signal by the personal computer allows it to communicate with the analog data generating and processing device as if it were a mass storage device even though it is not a mass storage device.

83. (new) The analog data generating and processing device of claim 82, wherein receipt and processing of the response signal by the personal computer allows it to communicate with the analog data generating and processing device as if it were a mass storage device having a rotatable storage disk even though it is not a mass storage device having a rotatable storage disk.

84. (new) The analog data generating and processing device of claim 83, wherein receipt and processing of the response signal by the personal computer allows it to communicate with the analog data generating and processing device as if it were a hard disk drive even though it is not a hard disk drive.

85. (new) The analog data generating and processing device of claim 72, wherein the memory of the analog data generating and processing device comprises a buffer memory.

86. (new) The analog data generating and processing device of claim 72, wherein the connecting device is adapted to be operatively connected to a SCSI interface of the personal computer.

87. (new) The analog data generating and processing device of claim 72, wherein the processor comprises a digital signal processor.

88. (new) The analog data generating and processing device of claim 72, wherein the digitized versions of the analog data is transferred to the personal computer in a format suitable for a mass storage device present in the personal computer.

89. (new) The analog data generating and processing device of claim 88 wherein the processor is adapted to create a root directory and virtual files in the memory which can be accessed by the personal computer.

90. (new) The analog data generating and processing device of claim 89, wherein at least one of the virtual files comprises a configuration file stored in the memory.

91. (new) The analog data generating and processing device of claim 90, wherein the configuration file allows a user to configure the analog data generating and processing device as being a specific mass storage device.

92. (new) The analog data generating and processing device of claim 91, wherein the configuration file allows a user to configure the analog data generating and processing device as being a specific hard disk drive.

93. (new) The analog data generating and processing device of claim 72, wherein a wire based connection is used to operatively connect the input/output port of the processor circuit to the multi-purpose interface of the personal computer.

REMARKS

Claims 1-16 have been cancelled without prejudice as to the subject matter claimed therein. New claims 17-93 are being submitted herewith for the Examiner's consideration.

An Information Disclosure statement is being filed herewith for the Examiner's consideration. It is respectfully submitted that the claims presented in this preliminary amendment are patentable over all of the prior art references included in the IDS, either taken along or in a purported combination, for a number of different reasons, including those that are discussed in greater detail hereinafter.

The Examiner is respectfully requested to review the following eight references in detail, all of which are listed in the IDS. Portions of each reference that one may argue allegedly are relevant to the subject matter of the currently pending claims, together with an identification of each reference, are presented hereinafter:

1) US Patent No. 5,915,106, which is entitled "Method And System For Operating A Scanner Which Emulates A Disk Drive," is not prior art to any of the claims submitted herewith. The earliest US filing date of this patent (March 20, 1997) is sixteen days after the earliest effective filing date of the currently pending claims, which is the March 4, 1997 filing date of German application no. 197 08 755. The Examiner's confirmation of this is earnestly solicited.

2) US Patent No. 5,508,821 is entitled "Image Scanner And Image Forming Apparatus With An Interface For Connection With An External Computer." Column 4, lines 21-23 of this patent state that the "image scanner 20 emulates the file system of 'UNIX' as if it were a hard disc. Accordingly, the image scanner 20

looks like the hard disc from the workstation 21 can be handled as a hard disk." In the summary of the invention of this patent, it is stated that an "object" of the invention is to provide an "image scanner" that "requires no preparation of any new device driver."

3) US Patent No. 5,131,089 is entitled "Solid State disk Drive Emulation." The abstract of this patent states that the "system permits software written for use with floppy disks to be used with solid state memory devices such as RAM cards or ROM without modification of the software."

4) US Patent No. 4,642,759 is entitled "Bubble Memory Disk Emulation."

5) A two page printout of text included with Windows 95 is submitted herewith concerning the "RAMDRIVE.SYS" command. This document states that this command allows a computer's RAM memory to simulate a hard disk drive.

6) Figure 1 of US Patent No. 5,724,574 discloses a hardware arrangement that includes, for example, a high speed scanner 24, a local area network 10, an optical disk based document server 15, and a number of workstations 18.

7) An article entitled "Optical Server Uses Network Protocol For Plug-And-Play Integration" was published in 1993. Page two of this article states that "emulation of the magnetic file system with a WORM-specific file system in this manner has several distinct advantages. The principal advantage is that the WORM disk appears to applications and utilities as just another disk."

8) The manual for Polaroid's Digital Camera model no. PDC-2000 indicates

that it was published in 1996. The Examiner is asked to assume, for the sake of argument, that this is the case. Applicant reserves the right to challenge this in all forums and proceedings other than the examination of this application.

Page 11 of the manual states that the "PDC-2000 camera is a Small Computer Systems Interface (SCSI) device," that one can "connect up to seven SCSI devices to your computer," and that the "PDC-2000 camera's SCSI ID is preset to 4 at the factor."

Page 83 of the manual states that to "transfer and work with pictures from the PDC-2000 camera on your PC, you use the PDC-2000 TWAIN driver . . ." or one can install "PDC-2000 Direct" software.

The currently pending claims clearly are supported by the specification as originally filed. As one example, all of the currently pending claims generally require that a sensor of some kind be adapted to be exposed to analog wave signals (*e.g.*, electromagnetic radiation) that originate at least in part from a source that is external to and not located in substantial proximity to the housing in which the sensor is contained. These claim features are supported, for example, by the "diagnostic radiology system" disclosed at page 1, paragraph 4, line 3 of the specification of the instant application.

An example of such a "diagnostic radiology system" is, for example, an x-ray machine, the x-rays being one example of the claimed "analog wave signals." As readily apparent to one of ordinary skill in the relevant art, typical x-ray machines include two housings – one in which an x-ray generator is mounted and a second one in which an x-ray transducer is mounted. The x-ray generator is physically separated from and not located in substantial proximity to the

transducer so that, for example, a patient can position his or her leg between the generator and the transducer. The transducer creates a set of analog data that comprises an x-ray so that, for example, a user can determine whether the patient's leg is broken.

It should be noted that the scope of the currently pending claims *is not* limited to "diagnostic radiology systems" and or to systems that only produce "x-rays." In this regard, other "modes" of practicing the claimed invention include, for example, the CCD device of a camera that is exposed to ambient light, and that creates therefrom a set of analog data representative of a picture. Further "modes" of practicing the claimed invention include, for example, a camera having a CCD that is adapted to receive ambient light as well as light from the camera's flash, and that creates therefrom a set of analog data representative of a picture. Other types of "sensors" within the scope of the present invention include, for example, dictaphone transducers that change analog voice signals into analog vocal signals.

For the Examiner's information, the inclusion of the above-described subject matter in the currently pending claims is one reason that the Examiner should find these claims patentable over, for example, the prior art of record that discloses the use of document scanners (*e.g.*, US Patent Nos. 5,508,821, 5,532,825 and 5,724,574). In contrast to the currently pending claims, the scanner references teach a light source that is located inside the scanner and that is located in substantial proximity to the CCD of the scanner. Such sensors *are not* adapted to process analog wave signals such as, for example, ambient light or other electromagnetic radiation that is present outside of the scanner housing or that is reflected off of or pass through objects that are not located in substantial proximity to the scanner. For this reason alone, the currently pending claims should be found to be patentable over the scanner references.

A second example of how the currently pending claims clearly are supported by the originally filed specification is as follows. One aspect of the currently pending claims is that analog data can be generated, digitized and stored in memory *irrespective* of whether communication with a personal computer already has been initiated. Support for this claim element is found, for example, at the third full paragraph on page 11 of the specification, which states:

> "As described above, a data buffer can be implemented in the memory means 14 to permit independence in terms of time of the data transmit/receive device attached to the second connecting device from the hose device attached to the first connecting device."

A still further aspect of the currently pending claims that is fully supported by the originally filed specification is as follows. All of the claims presented in this preliminary amendment generally require that the analog data generating and processing device (*e.g.*, an x-ray machine or a digital camera having a flash) be operatively connected to a multi-purpose interface of a personal computer (by, for example, a wire-based connection), and further be *automatically and without user intervention* recognized by the personal computer as being a device that has digital data which is stored therein and which is selectively retrievable therefrom. Support for these claim features is found, for example, at page 7, lines 8-12 of the specification, which recite:

> "The digital signal processor 13 receives this inquiry instruction via the first connecting device and generates a signal which is sent to the host device (not shown) again via the first connecting device 12 and the host line 11. This signal

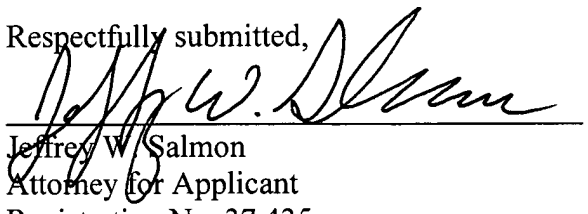
indicates to the host device that, for example, a hard disk drive is attached at the interface to which the INQUIRY instruction was sent."

It is respectfully submitted that no prior art reference of record, either taken alone or in a purported combination, teaches or suggests the combinations claimed in the currently pending claims for a number of different reasons. As one example, US Patent No. 5,508,821 does not teach or suggest, for example, the above-noted "automatic recognition" feature because, for example, the system disclosed therein is UNIX based. As readily apparent to one of ordinary skill in the relevant art, such UNIX based systems affirmatively require user intervention in order to operate and use the scanner disclosed in the '821 patent.

As a further example of the patentability of the currently pending claims, the camera disclosed in the Polaroid manual submitted (assuming, for argument's sake that it is prior art) cannot be automatically recognized without human intervention. In this regard, user intervention always is required because, for example, a user needs to make sure that the camera's SCSI identification number does not conflict with the ID number of any other device in a daisy chain of which the camera forms a part. For this reason alone, for example, the currently pending claims should be found to be patentable over the Polaroid camera manual (assuming for argument's sake that it is prior art).

It is respectfully submitted that the new claims are in condition for allowance and, therefore, a formal notice to that effect is earnestly solicited. In this regard, the Examiner is respectfully requested to contact the undersigned attorney upon entry of this amendment.

Respectfully submitted,



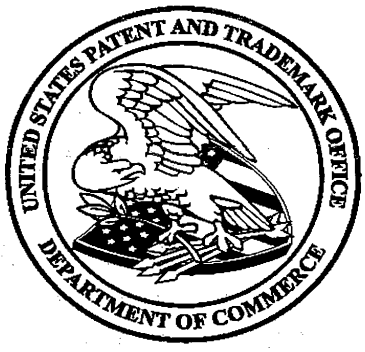
Jeffrey W. Salmon
Attorney for Applicant
Registration No. 37,435

March 28, 2006
Welsh & Katz, Ltd.
120 South Riverside Plaza
22nd Floor
Chicago, IL 60606
Telephone (312) 655-1500
Facsimile (312) 655-1501

Exhibit J

Manual of PATENT EXAMINING PROCEDURE

Original Eighth Edition, August 2001
Latest Revision September 2007



U.S. DEPARTMENT OF COMMERCE
United States Patent and Trademark Office

The U.S. Patent and Trademark Office does not handle the sale of the Manual, distribution of notices and revisions, or change of address of those on the subscription list. Correspondence relating to existing subscriptions should be sent to the Superintendent of Documents at the following address:

Superintendent of Documents
Mail List Section
Washington, DC 20402
Telephone: 202-512-2267

Inquiries relating to purchasing the Manual should be directed to:

Superintendent of Documents
United States Government Printing Office
Washington, DC 20402
Telephone: 202-512-1800

Orders for reproduced copies of individual replacement pages or of previous revisions of the Manual should be sent to the following address:

Mail Stop Document Services
Director of the U.S. Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
Telephone: 1-800-972-6382 or 571-272-3150

Previous editions and revisions of the Manual are available on microfilm in the Patent Search Room.
The Manual is available on CD-ROM and on diskette from:

U.S. Patent and Trademark Office
Office of Electronic Information Products
MDW 4C18, P.O. Box 1450
Alexandria, VA 22313-1450
Telephone: 571-272-5600

Employees of the U.S. Patent and Trademark Office should direct their requests for the Manual, replacement pages, notices, and revisions to the Office of Patent Training. Telephone: 571-272-7222

Pursuant to the Patent and Trademark Office Efficiency Act (PTOEA) (Pub. L. 106-113, 113 Stat. 1501A-572), the head of the United States Patent and Trademark Office (USPTO) is the "Under Secretary of Commerce for Intellectual Property and Director of the United States Patent and Trademark Office." The Director is assisted by the "Deputy Under Secretary of Commerce for Intellectual Property and Deputy Director of the United States Patent and Trademark Office." The patent operations of the USPTO are now headed by the "Commissioner for Patents." The trademark operations of the USPTO are now headed by the "Commissioner for Trademarks." Under section 4741(b) of the PTOEA, any reference to the Commissioner of Patents and Trademarks, the Assistant Commissioner for Patents, or the Assistant Commissioner for Trademarks is deemed to refer to the Director, the Commissioner for Patents, or the Commissioner for Trademarks, respectively. See "Reestablishment of the Patent and Trademark Office as the United States Patent and Trademark Office" published in the *Federal Register* at 65 FR 17858 (Apr. 5, 2000), and in the *Official Gazette of the United States Patent and Trademark Office* at 1234 O.G. 41 (May 9, 2000).

Additions to the text of the Manual are indicated by arrows (><) inserted in the text. Deletions are indicated by a single asterisk (*) where a single word was deleted and by two asterisks (**) where more than one word was deleted. The use of three or five asterisks in the body of the laws, rules, treaties, and administrative instructions indicates a portion of the law, rule, treaty, or administrative instruction which was not reproduced.

First Edition, November 1949
Second Edition, November 1953
Third Edition, November 1961
Fourth Edition, June 1979
Fifth Edition, August 1983
Sixth Edition, January 1995
Seventh Edition, July 1998
Eighth Edition, August 2001
Revision 1, February 2003
Revision 2, May 2004
Revision 3, August 2005
Revision 4, October 2005
Revision 5, August 2006
Revision 6, September 2007

made. The examiner should analyze whether the metes and bounds of the claim are clearly set forth. Examples of claim language which have been held to be indefinite because the intended scope of the claim was unclear are:

(A) "R is halogen, for example, chlorine";

(B) "material such as rock wool or asbestos" *Ex parte Hall*, 83 USPQ 38 (Bd. App. 1949);

(C) "lighter hydrocarbons, such, for example, as the vapors or gas produced" *Ex parte Hasche*, 86 USPQ 481 (Bd. App. 1949); and

(D) "normal operating conditions such as while in the container of a proportioner" *Ex parte Steigerwald*, 131 USPQ 74 (Bd. App. 1961).

>The above examples of claim language which have been held to be indefinite are fact specific and should not be applied as *per se* rules. See MPEP § 2173.02 for guidance regarding when it is appropriate to make a rejection under 35 U.S.C. 112, second paragraph.<

2173.05(e) Lack of Antecedent Basis [R-5]

A claim is indefinite when it contains words or phrases whose meaning is unclear. The lack of clarity could arise where a claim refers to "said lever" or "the lever," where the claim contains no earlier recitation or limitation of a lever and where it would be unclear as to what element the limitation was making reference. Similarly, if two different levers are recited earlier in the claim, the recitation of "said lever" in the same or subsequent claim would be unclear where it is uncertain which of the two levers was intended. A claim which refers to "said aluminum lever," but recites only "a lever" earlier in the claim, is indefinite because it is uncertain as to the lever to which reference is made. Obviously, however, the failure to provide explicit antecedent basis for terms does not always render a claim indefinite. If the scope of a claim would be reasonably ascertainable by those skilled in the art, then the claim is not indefinite. *>Energizer Holdings Inc. v. Int'l Trade Comm'n, 435 F.3d 1366, 77 USPQ2d 1625 (Fed. Cir. 2006)(holding that "anode gel" provided by implication the antecedent basis for "zinc anode");< Ex parte Porter*, 25 USPQ2d 1144, 1145 (Bd. Pat. App. & Inter. 1992) ("controlled stream of fluid" provided reasonable antecedent basis for "the controlled fluid"). Inherent components of elements recited have antecedent basis in the recitation of the components themselves. For example, the limitation "the outer surface of said sphere" would not require an antecedent recitation that the sphere has an outer surface. See *Bose Corp. v. JBL, Inc.*, 274 F.3d 1354, 1359, 61 USPQ2d 1216, 1218-19 (Fed. Cir 2001) (holding that recitation of "an ellipse" provided antecedent basis for "an ellipse having a major diameter" because "[t]here can be no dispute that mathematically an inherent characteristic of an ellipse is a major diameter").

EXAMINER SHOULD SUGGEST CORRECTIONS TO ANTECEDENT PROBLEMS

Antecedent problems in the claims are typically drafting oversights that are easily corrected once they are brought to the attention of applicant. The examiner's task of making sure the claim language complies with the requirements of the statute should be carried out in a positive and constructive way, so that minor problems can be identified and easily corrected, and so that the major effort is expended on more substantive issues. However, even though indefiniteness in claim language is of semantic origin, it is not rendered unobjectionable simply because it could have been corrected. *In re Hammack*, 427 F.2d 1384 n.5, 166 USPQ 209 n.5 (CCPA 1970).

A CLAIM TERM WHICH HAS NO ANTECEDENT BASIS IN THE DISCLOSURE IS NOT NECESSARILY INDEFINITE

The mere fact that a term or phrase used in the claim has no antecedent basis in the specification disclosure does not mean, necessarily, that the term or phrase is indefinite. There is no requirement that the words in the claim must match those used in the specification disclosure. Applicants are given a great deal of latitude in how they choose to define their invention so long as the terms and phrases used define the invention with a reasonable degree of clarity and precision.

A CLAIM IS NOT *PER SE* INDEFINITE IF THE BODY OF THE CLAIM RECITES ADDITIONAL ELEMENTS WHICH DO NOT APPEAR IN THE PREAMBLE

The mere fact that the body of a claim recites additional elements which do not appear in the claim's

preamble does not render the claim indefinite under 35 U.S.C. 112, second paragraph. See *In re Larsen*, No. 01-1092 (Fed. Cir. May 9, 2001) (unpublished) (The preamble of the *Larsen* claim recited only a hanger and a loop but the body of the claim positively recited a linear member. The examiner rejected the claim under 35 U.S.C. 112, second paragraph, because the omission from the claim's preamble of a critical element (i.e., a linear member) renders that claim indefinite. The court reversed the examiner's rejection and stated that the totality of all the limitations of the claim and their interaction with each other must be considered to ascertain the inventor's contribution to the art. Upon review of the claim in its entirety, the court concluded that the claim at issue apprises one of ordinary skill in the art of its scope and, therefore, serves the notice function required by 35 U.S.C. 112, paragraph 2.).

2173.05(f) Reference to Limitations in Another Claim

A claim which makes reference to a preceding claim to define a limitation is an acceptable claim construction which should not necessarily be rejected as improper or confusing under 35 U.S.C. 112, second paragraph. For example, claims which read: "The product produced by the method of claim 1." or "A method of producing ethanol comprising contacting amylose with the culture of claim 1 under the following conditions ....." are not indefinite under 35 U.S.C. 112, second paragraph, merely because of the reference to another claim. See also *Ex parte Porter*, 25 USPQ2d 1144 (Bd. Pat. App. & Inter. 1992) where reference to "the nozzle of claim 7" in a method claim was held to comply with 35 U.S.C. 112, second paragraph. However, where the format of making reference to limitations recited in another claim results in confusion, then a rejection would be proper under 35 U.S.C. 112, second paragraph.

2173.05(g) Functional Limitations [R-3]

A functional limitation is an attempt to define something by what it does, rather than by what it is (e.g., as evidenced by its specific structure or specific ingredients). There is nothing inherently wrong with defining some part of an invention in functional terms. Functional language does not, in and of itself, render a claim improper. *In re Swinehart*, 439 F.2d 210, 169 USPQ 226 (CCPA 1971).

A functional limitation must be evaluated and considered, just like any other limitation of the claim, for what it fairly conveys to a person of ordinary skill in the pertinent art in the context in which it is used. A functional limitation is often used in association with an element, ingredient, or step of a process to define a particular capability or purpose that is served by the recited element, ingredient or step. >In *Innova/Pure Water Inc. v. Safari Water Filtration Sys. Inc.*, 381 F.3d 1111, 1117-20, 72 USPQ2d 1001, 1006-08 (Fed. Cir. 2004), the court noted that the claim term "operatively connected" is "a general descriptive claim term frequently used in patent drafting to reflect a functional relationship between claimed components," that is, the term "means the claimed components must be connected in a way to perform a designated function." "In the absence of modifiers, general descriptive terms are typically construed as having their full meaning." *Id.* at 1118, 72 USPQ2d at 1006. In the patent claim at issue, "subject to any clear and unmistakable disavowal of claim scope, the term 'operatively connected' takes the full breath of its ordinary meaning, i.e., 'said tube [is] operatively connected to said cap' when the tube and cap are arranged in a manner capable of performing the function of filtering." *Id.* at 1120, 72 USPQ2d at 1008.<

Whether or not the functional limitation complies with 35 U.S.C. 112, second paragraph, is a different issue from whether the limitation is properly supported under 35 U.S.C. 112, first paragraph, or is distinguished over the prior art. A few examples are set forth below to illustrate situations where the issue of whether a functional limitation complies with 35 U.S.C. 112, second paragraph, was considered.

It was held that the limitation used to define a radical on a chemical compound as "incapable of forming a dye with said oxidizing developing agent" although functional, was perfectly acceptable because it set definite boundaries on the patent protection sought. *In re Barr*, 444 F.2d 588, 170 USPQ 33 (CCPA 1971).

In a claim that was directed to a kit of component parts capable of being assembled, the Court held that limitations such as "members adapted to be positioned" and "portions . . . being resiliently dilatable whereby said housing may be slidably positioned" serve to precisely define present structural attributes

Exhibit K

Westlaw.

H

Ampex Corp. v. Eastman Kodak Co.
D.Del.,2006.

United States District Court,D. Delaware.
AMPEX CORPORATION, Plaintiff,
v.
EASTMAN KODAK COMPANY, and Altek Corporation, Defendants.
No. CIV.A. 04-1373-KAJ.

Oct. 31, 2006.

Background: Patentee brought action against competitors, alleging infringement of its patent for a digital camera. Competitors moved for summary judgment.

Holdings: The District Court, Jordan, J., held that:
(1) patent was not literally infringed, and
(2) patent was not infringed under the doctrine of equivalents.

Motion granted.

West Headnotes

[1] Patents 291 ⚿235(2)

291 Patents
291XII Infringement
291XII(A) What Constitutes Infringement
291k233 Patents for Machines or Manufactures
291k235 Identity of Principle or Mode of Operation
291k235(2) k. Particular Patents or Devices. Most Cited Cases
Competitors' digital cameras did not literally infringe claim in camera patent requiring full video pixel data comprising a full size image to be stored in bulk storage memory; claim required that the pixel values not change before storage in permanent memory, and numeric values representing at least some of the pixels in an image were changed before storage in permanent memory of competitors' cameras.

[2] Patents 291 ⚿168(3)

291 Patents
291IX Construction and Operation of Letters Patent
291IX(B) Limitation of Claims
291k168 Proceedings in Patent Office in General
291k168(3) k. Rejection and Amendment of Claims of Particular Patents. Most Cited Cases
Prosecution history estoppel barred patentee's claim that competitor's digital camera data storage system infringed its patent under the doctrine of equivalents; patentee amended its patent application to require video pixel data to be unchanged before storage in permanent memory to avoid indefiniteness rejection, even though it was foreseeable that the pixel values of an image could be altered, in some fashion, prior to storage in the permanent memory of a digital device, and still represent a substantially similar image. 35 U.S.C.A. § 112.

[3] Patents 291 ⚿312(1.6)

291 Patents
291XII Infringement
291XII(C) Suits in Equity
291k312 Evidence
291k312(1) Presumptions and Burden of Proof
291k312(1.6) k. Laches and Estoppel. Most Cited Cases
Patentee's narrowing amendment of claim limitation to overcome a rejection on ground of indefiniteness raised presumption that the patentee surrendered all territory between the original claim limitation and the amended claim limitation. 35 U.S.C.A. § 112.

[4] Patents 291 ⚿312(1.6)

© 2008 Thomson Reuters/West. No Claim to Orig. U.S. Govt. Works.

291 Patents
291XII Infringement
291XII(C) Suits in Equity
291k312 Evidence
291k312(1) Presumptions and Burden of Proof
291k312(1.6) k. Laches and Estoppel. Most Cited Cases
Patentee may rebut presumption that it has surrendered all territory between the original claim limitation and the amended claim limitation if it can demonstrate that the alleged equivalent would have been unforeseeable at the time of the narrowing amendment, that the rationale underlying the narrowing amendment bore no more than a tangential relation to the equivalent in question, or that there was some other reason suggesting that the patentee could not reasonably have been expected to have described the alleged equivalent. 35 U.S.C.A. § 112.

Patents 291 ⚿328(2)

291 Patents
291XIII Decisions on the Validity, Construction, and Infringement of Particular Patents
291k328 Patents Enumerated
291k328(2) k. Original Utility. Most Cited Cases
4,821,121. Not Infringed.

***564** Jack B. Blumenfeld, Julie Heaney, Morris, Nichols, Arsht & Tunnell LLP, Wilmington, DE, Ropes & Gray LLP, New York, NY (Jesse J. Jenner, Sasha G. Rao, of counsel), Ropes & Gray LLP, Palo Alto, CA (Norman H. Beamer, Gabrielle E. Higgins, of counsel), Ropes & Gray LLP, Washington, DC (James E. Hopenfeld, of counsel), Counsel for Plaintiff.
Collin J. Seitz, Jr., Connolly Bove Lodge & Hutz, LLP, Wilmington, DE, Wilmer Cutler Pickering Hale and Dorr LLP, Boston, MA (William F. Lee, Donald R. Steinberg, Esq., Michael J. Summersgill, of counbsel), S. Calvin Walden, Wilmer Cutler Pickering Hale and Dorr LLP, New York, NY, Counsel for Defendants.

MEMORANDUM OPINION

JORDAN, District Judge.

I. INTRODUCTION

This patent infringement case is before me on the motion of Defendants, Eastman Kodak Company and Altek Corporation (collectively, "Defendants"), for summary judgment of non-infringement. (Docket Item ["D.I."] 302; the "Motion".) The background of this dispute is set forth in the claim construction opinion I issued on October 26, 2006. (D.I.472.) For the reasons set forth herein, I will grant Defendants' Motion.

II. STANDARD OF REVIEW

Pursuant to Federal Rule of Civil Procedure 56(c), a party is entitled to summary judgment if a court determines from its examination of "the pleadings, depositions, answers to interrogatories, and admissions on file, together with the affidavits, if any," that there are no genuine issues of material fact and that the moving party is entitled to judgment as a matter of law. Fed.R.Civ.P. 56(c). In determining whether there is a genuine issue of material fact, a court must review the evidence and construe all inferences in the light most favorable to the non-moving party. *Goodman v. Mead Johnson & Co.*, 534 F.2d 566, 573 (3d Cir.1976). However, a court should not make credibility determinations or weigh the evidence. *Reeves v. Sanderson Plumbing Prods., Inc.*, 530 U.S. 133, 150, 120 S.Ct. 2097, 147 L.Ed.2d 105 (2000). To defeat a motion for summary judgment, the non-moving party must "do more than simply show that there is some metaphysical doubt as to the material facts." *Matsushita Elec. Indus. Co., Ltd. v. Zenith Radio Corp.*, 475 U.S. 574, 586-87, 106 S.Ct. 1348, 89 L.Ed.2d 538 (1986) (internal citation omitted). The non-moving party "must set forth specific facts showing that there is a genuine issue for trial." Fed.R.Civ.P. 56(c). "Where the record taken as a whole could not lead a rational trier of fact to find for the non-

© 2008 Thomson Reuters/West. No Claim to Orig. U.S. Govt. Works.

moving party, there is no genuine issue for trial." *Matsushita*, 475 U.S. at 587, 106 S.Ct. 1348 (internal citation omitted). Accordingly, a mere scintilla of evidence in support of the non-moving party is insufficient for a court to deny summary judgment. *Anderson v. Liberty Lobby, Inc.*, 477 U.S. 242, 252, 106 S.Ct. 2505, 91 L.Ed.2d 202 (1986).

III. DISCUSSION

A. Non-Infringement

Defendants have moved for summary judgment that the accused digital cameras do not infringe asserted claims 7, 8, and 10-15 of U.S. Patent No. 4,821,121 (the " '121 patent"). (D.I.302.) A patent infringement analysis involves two steps: claim construction and then the application of the construed claim to the accused process or product. *Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 976 (Fed.Cir.1995) (en banc), *aff'd*,517 U.S. 370, 116 S.Ct. 1384, 134 L.Ed.2d 577 (1996). The first step, claim construction, has been held to be purely a matter of law. *See* ***565***Cybor Corp. v. FAS Techs., Inc.*, 138 F.3d 1448, 1454 (Fed.Cir.1998) (en banc). The second step, application of the claim to the accused product, is a fact-specific inquiry. *See Kustom Signals, Inc. v. Applied Concepts, Inc.*, 264 F.3d 1326, 1332 (Fed.Cir.2001) (Patent infringement, "whether literal or under the doctrine of equivalents, is a question of fact."). Summary judgment is appropriate in patent infringement suits when it is apparent that only one conclusion regarding infringement could be reached by a reasonable jury. *See Telemac Cellular Corp. v. Topp Telecom, Inc.*, 247 F.3d 1316, 1323 (Fed.Cir.2001). Having completed the claim construction step (D.I.472), I now consider the parties' arguments regarding infringement.

1. Literal Infringement

[1] To prevail on a motion for summary judgment of non-infringement, Defendants must prove that the accused products do not practice at least one of the claim limitations in each asserted claim. Defendants rely on several limitations in the claims of the '121 patent to support their argument for non-infringement. First, Defendants argue that their digital cameras do not capture or receive a "video image" that comes from, or forms a part of, a series of related electronic images created for rapid display to allow the appearance of movement. (D.I. 304 at 22.) Second, Defendants claim that their cameras do not store, for later access, the same "data" that was initially received and used to generate a reduced size image. (*Id.* at 27.) Third, according to Defendants, the accused cameras do not transfer data "directly" between the random access memory and bulk memory because there is intervening circuitry. (*Id.* at 34.) Next, Defendants contend that, in their cameras, the random access memory does not contain "an input port and an output port" because it has a single port that performs both the input and output functions. (*Id.* at 37.) Finally, Defendants argue that the accused cameras obtain images from a source located completely within the housing of the camera, and not from an "external source." (*Id.* at 38.) Because Defendants' second argument, regarding the "data" that is stored for later access, is entirely dispositive, I consider only it.

Each asserted claim uses the term "video data," "video pixel data," "data set," or "image data set" to initially identify the data representing an image. ('121 patent at 6:27-31, 6:49-55, 7:36-41, 7:65-68, 8:16-18, 8:48-51, 8:65-9:3, 10:8-13.) When later describing the storage of that data in permanent memory, each claim, except claim 11,FN1 refers to the data using the term "said" or "the." (*Id.* at 6:32-33, 6:60-61, 7:50-51, 8:25-27, 8:56-59, 9:8-11, 10:19-22.) For example, claim 7 recites a "random access memory means for storing video pixel data" and a "bulk memory means for receiving *said* video pixel data from said random access memory means." (*Id.* at 6:23-40 (emphasis added).) As set forth in my claim construction opinion, all references to data in the form "said/the data" mean "numerical information representing the

© 2008 Thomson Reuters/West. No Claim to Orig. U.S. Govt. Works.

same luminance, red chrominance, and blue chrominance components of each pixel in a video image." (D.I. 472 at 39.) Therefore, Defendants correctly assert that the data stored in the bulk (permanent) memory must represent the same pixel values as the data that was originally received by the still store system and stored in the random access memory.

> FN1. Claim 11 does not require storage of "said/the data," but rather storage of "the full size image and the reduced size image in bulk storage memory."('121 patent at 8:9-10.) As a result, claim 11 will be discussed separately.

Ampex asserts that Defendants' argument does not apply to claim 11 because it does not state that "said data" or "the data" is stored in the bulk memory. (D.I. ***566** 366 at 28; D.I. 477.) However, claim 11 does recite "receiving and storing ... in a random access memory, full video pixel data comprising a full size image" and then "storing ...*the* full size image ... in bulk storage memory."('121 patent at 7:65-68, 8:9-10 (emphasis added).) In this claim, "*the* full size image" indicates that the same full size image is stored in the random access memory and the bulk memory. As I stated in my claim construction opinion, "in order for digital data to represent the *same* image, it must represent the *same* luminance, red chrominance, and blue chrominance values for each pixel of that image." (D.I. 472 at 13-14 (original emphasis).) This means that, in order to store the same full size image in both the random access memory and the bulk memory, the data in each memory must represent the same pixel values. Therefore, claim 11, like the other claims asserted by Ampex, requires that the pixel values do not change before storage in permanent memory.

Defendants argue that their cameras cannot satisfy the "data" limitation because their cameras perform significant processing which changes the pixel values of an image before storage in permanent memory. (D.I. 304 at 28-31.) In particular, Defendants claim that pixel values are altered when the accused cameras perform CFA interpolation, white balancing, color correction, tone correction, edge correction, and JPEG compression. (*Id.;* D.I. 409 at 8.) Ampex does not dispute that Defendants' cameras perform these processing steps before storing the data. (D.I. 366 at 15-16, 28-32.) And, Ampex admits that "the numerical values stored for a given pixel in a taken picture change because of processing in the camera." (*Id.* at 29; *see also id.* at 16 (these image processing steps ... may change the numeric value of pixels').) Ampex's expert, Dr. Ligler, confirms that, at a minimum, the processing performed by Defendants' cameras changes the value of some of the pixels in an image. (D.I. 306 at A-496, 258:7-17 (agreeing that CFA interpolation would change the mathematical values used to represent a pixel); *id.* at A-496, 258:18-20 (admitting that white balancing can alter pixels); *id.* at A-497-98, 259:9-260:3 (stating that color correction will alter at least some of the pixel values that represent an image); *id.* at A-498-99, 260:15-261:7 (admitting that tone correction would alter the mathematical value of pixels); *id.* at A-499-500, 261:16-262:9 (stating that edge correction would change the numeric value of pixels on or near an edge); *id.* at A-501, 263:7-16 (admitting that the mathematical value of pixels would be different after JPEG compression).) Since the numeric values representing at least some of the pixels in an image are changed before storage in permanent memory, Defendants' cameras cannot literally infringe any of the claims asserted by Ampex.

2. Doctrine of Equivalents

[2] Defendants argue that Ampex is not entitled to any scope of equivalents with respect to the claim limitations "said data" and "the data." (D.I. 304 at 32.) Specifically, Defendants point to an amendment in which "said" and "the" were added to several of the claims. (D.I. 306 at A-113-14, A-116.) Immediately prior to this amendment, the examiner had rejected the claims under 35 U.S.C. § 112 on numerous grounds, including that, "[i]n claim 18,

© 2008 Thomson Reuters/West. No Claim to Orig. U.S. Govt. Works.

line 11, 'video pixel data' is indefinite because it is not clear if it refers back to the pixel data recited in lines 5 and 6." FN2 ***567** (*Id.* at A-103.) In response, the term "said" was inserted into that claim so that it read "memory means for receiving *said* video pixel data from said random access memory means." FN3 (*Id.* at A-113 (emphasis added).) The applicant explained that the claims were amended to provide antecedents for the terms that the examiner identified as being indefinite. (*Id.* at A-123.) Specifically, the applicant stated that now the "video pixel data" in line 11 properly refers back to "video pixel data" in lines 5 and 6. (*Id.*) Defendants contend that this amendment raises the presumption of prosecution history estoppel. (D.I. 304 at 32.)

> FN2. The claim referred to as claim 18 during prosecution ultimately issued as claim 7 of the '121 patent. (D.I. 306 at A-135, A-152.)
>
> FN3. In the same amendment, claim 19 was amended to recite "bulk storage memory for also storing *the* video pixel data" (D.I. 306 at A-114), and claim 23 was amended to read "a second store for receiving and storing both *the* video data from the first store" (*id.* at A-116).

[3] "The first question in a prosecution history estoppel inquiry is whether an amendment ... has narrowed the literal scope of a claim." *Festo Corp. v. Shoketsu Kinzoku Kogyo Kabushiki Co.*, 344 F.3d 1359, 1366 (Fed.Cir.2003) (citing *Pioneer Magnetics, Inc. v. Micro Linear Corp.*, 330 F.3d 1352, 1356 (Fed.Cir.2003)). In this case, the term "video pixel data," by itself, means numerical information representing the pixel values of an image. By adding the term "said" to "video pixel data," the amendment narrowed the scope of this term. The applicant specifically stated that the purpose of the amendment was to limit the term "video pixel data" to the data that was identified earlier in the same claim. Since the amendment at issue was narrowing, the next question is "whether the reason for that amendment was a substantial one relating to patentability." *Id.* The U.S. Supreme Court expressly recognized that "a narrowing amendment made to comply with any provision of the Patent Act, including § 112, may invoke an estoppel." *Id.* (citing *Festo Corp. v. Shoketsu Kinzoku Kogyo Kabushiki Co.*, 535 U.S. 722, 736, 122 S.Ct. 1831, 152 L.Ed.2d 944 (2002) ("*Festo VIII* ")). Since the applicant here made a narrowing amendment to overcome a rejection under section 112, there is a "presumption that the patentee has surrendered all territory between the original claim limitation and the amended claim limitation." *Id.* at 1367 (citing *Festo VIII*, 535 U.S. at 740, 122 S.Ct. 1831).

[4] Ampex may rebut that presumption if it can demonstrate "that the alleged equivalent would have been unforeseeable at the time of the narrowing amendment, that the rationale underlying the narrowing amendment bore no more than a tangential relation to the equivalent in question, or that there was 'some other reason' suggesting that the patentee could not reasonably have been expected to have described the alleged equivalent." *Id.* at 1368 (citing *Festo VIII*, 535 U.S. at 740-41, 122 S.Ct. 1831). Ampex argues that the alleged equivalent in this case would have been unforeseeable at the time of the narrowing amendment. (D.I. 366 at 33.) In particular, Ampex claims that, although the image processing performed by Defendants' cameras was well known at the time the '121 patent was filed, it was unforeseeable that progress of component miniaturization would allow the video input circuit (i.e., the CCD image sensor) to be integrated into the same chassis as the circuitry used for image storage. (*Id.* at 30.) It was this advance in technology, says Ampex, that permitted image processing to occur between the random access memory and the bulk memory. (*Id.*) While Defendants contest the accuracy of that argument (D.I. 409 at 13), Ampex has not shown that the alleged equivalent was unforeseeable***568** at the time of the amendment, even if its argument were accepted as true.

Ampex's position is that, while the image pro-

© 2008 Thomson Reuters/West. No Claim to Orig. U.S. Govt. Works.

cessing performed by Defendants' cameras may change some pixel values prior to storage, the data that is ultimately stored in Defendants' cameras will represent an image that is equivalent to the image first captured and stored. (D.I. 366 at 30-31.) That position has intuitive appeal, but it ignores the import of *Festo* and following cases. Ampex's argument regarding the miniaturization of components, if true,[FN4] only proves that it was unforeseeable that the particular image processing techniques used by Defendants' cameras could be performed between the random access memory and the bulk memory. It certainly does not prove that it was unforeseeable at the time Ampex amended its application that the pixel values of an image could be altered, in some fashion, prior to storage in the permanent memory of a digital device, and still represent a substantially similar image. Therefore, it is reasonable to expect the patentee to have described a still store system in which the pixel values of an image may change before storage. But, instead, the claims of the '121 patent were specifically amended to indicate that the claimed system did not alter the pixel values of an image. (D.I. 306 at A-123.)

> FN4. I too harbor doubts that it was unforeseeable that advances in miniaturization technology would permit the design of devices like the Defendants', but I need not reach that issue.

Ampex also contends that the rationale underlying the amendment bears no more than a tangential relation to the equivalent in question. (D.I. 366 at 33.) In assessing this argument, the question is essentially "whether the reason for the amendment is peripheral, or not directly relevant, to the alleged equivalent." *Festo,* 344 F.3d at 1369 (citing *The American Heritage College Dictionary* 1385 (3d ed.1997); 2 *The New Shorter Oxford English Dictionary* 3215-16 (1993)). In this case, it would be difficult to imagine an amendment that is more directly related to the equivalent at issue. As discussed above, the prosecution history shows that the term "said" was added for the purpose of indicating an antecedent relationship between the pixel data first stored and the data subsequently referenced, or, more specifically, to demonstrate that the data stored in the bulk memory of the still store system is the same as that stored in the random access memory. (D.I. 306 at A-123.) Given that the amendment specifies exactly what data will be stored by the claimed invention, it is directly relevant to the alleged equivalent.

Since Ampex has failed to rebut the *Festo* presumption, prosecution history estoppel prevents the doctrine of equivalents from being applied to the "said data" element in the amended claims. *See Festo,* 344 F.3d at 1367. Furthermore, any "subject matter surrendered via claim amendments during prosecution is also relinquished for other claims containing the same limitation." *Glaxo Wellcome, Inc. v. Impax Labs., Inc.,* 356 F.3d 1348, 1356 (Fed.Cir.2004) (citing *Builders Concrete, Inc. v. Bremerton Concrete Prods. Co.,* 757 F.2d 255, 260 (Fed.Cir.1985)). In other words, prosecution history estoppel applies to the "said/the data" limitation in every claim asserted by Ampex regardless of whether it was ever amended. *See id.* As a result, Defendants' cameras cannot infringe any of the claims asserted by Ampex under the doctrine of equivalents.[FN5] Thus, there is no infringement, literally or by equivalents, of the "data" element that ***569** appears in every claim at issue in this case, and accordingly, I will grant Defendants' motion for summary judgment of non-infringement.

> FN5. Since the language "the full size image" imposes the same limitation as "said/the data" (*see supra* at 565), prosecution history estoppel also applies to that element of claim 11.

IV. CONCLUSION

Accordingly, I will grant Defendants' Motion for Summary Judgment of Non-Infringement. An appropriate order will follow.

© 2008 Thomson Reuters/West. No Claim to Orig. U.S. Govt. Works.

460 F.Supp.2d 563
460 F.Supp.2d 563
460 F.Supp.2d 563

ORDER

For the reasons set forth in the Memorandum Opinion issued in this matter today,

IT IS HEREBY ORDERED that the Defendants' Motion for Summary Judgment of Non-Infringement (D.I.302) is GRANTED.

D.Del.,2006.
Ampex Corp. v. Eastman Kodak Co.
460 F.Supp.2d 563

END OF DOCUMENT

© 2008 Thomson Reuters/West. No Claim to Orig. U.S. Govt. Works.

Exhibit L

SL #9/C

Notice of Allowability	Application No. 09/331,002	Applicant(s) TASLER, MICHAEL	
	Examiner Thuan N. Du	Art Unit 2185	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *amendment filed on March 18, 2002*.
2. ☒ The allowed claim(s) is/are *1-4 and 6-16 (renumbered as 1-15)*.
3. ☒ The drawings filed on *18 March 2002* are accepted by the Examiner.
4. ☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☒ All b) ☐ Some* c) ☐ None of the:
      1. ☒ Certified copies of the priority documents have been received.
      2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
      3. ☒ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.
5. ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).
   (a) ☐ The translation of the foreign language provisional application has been received.
6. ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application. **THIS THREE-MONTH PERIOD IS NOT EXTENDABLE**

7. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.

8. ☐ CORRECTED DRAWINGS must be submitted.
   (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
      1) ☐ hereto or 2) ☐ to Paper No. ____.
   (b) ☐ including changes required by the proposed drawing correction filed _____, which has been approved by the Examiner.
   (c) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No. _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the top margin (not the back) of each sheet. The drawings should be filed as a separate paper with a transmittal letter addressed to the Official Draftsperson.

9. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1☐ Notice of References Cited (PTO-892)
2☐ Notice of Informal Patent Application (PTO-152)
3☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
4☐ Interview Summary (PTO-413), Paper No.____ .
5☐ Information Disclosure Statements (PTO-1449), Paper No. ____.
6☒ Examiner's Amendment/Comment
7☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
8☐ Examiner's Statement of Reasons for Allowance
9☐ Other

THOMAS LEE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

#9/C

EXAMINER'S AMENDMENT

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

2. The application has been amended as follows:

3. Pursuant to MPEP 606.01, the title has been changed to read:

-- FLEXIBLE INTERFACE FOR COMMUNICATION BETWEEN A HOST AND AN ANALOG I/O DEVICE CONNTECTED TO THE INTERFACE REGARDLESS THE TYPE OF THE I/O DEVICE --

Conclusion

4. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Thuan N. Du whose telephone number is (703) 308-6292 or via e-mail, ***thuan.du@uspto.gov***. The examiner can normally be reached on Monday-Friday: 9:00 AM - 5:30 PM, EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Thomas C. Lee can be reached on (703) 305-9717.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Any response to this action should be mailed to:

Commissioner of Patents and Trademarks
Washington, D.C. 20231.

The fax numbers for the organization where this application or proceeding is assigned are as follow:

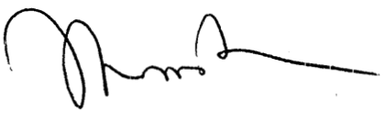
- (703) 746-7238 [After Final Communication]
- (703) 746-7239 [Official Communication]
- (703) 746-7240 [Non-Official Communication]

and/or:

(703) 746-5668 (use this fax number, only after approval by Examiner, for "INFORMAL" or "DRAFT" communication).

Hand-delivered responses should be brought to:

Crystal Park II
2121 Crystal Drive
Arlington, VA 22202
Fourth Floor (Receptionist).



THOMAS LEE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Thuan N. Du
May 15, 2002

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

24283 7590 05/17/2002

PATTON BOGGS
PO BOX 270930
LOUISVILLE, CO 80027

EXAMINER	
DU, THUAN N	
ART UNIT	CLASS-SUBCLASS
2185	710-001000

DATE MAILED: 05/17/2002

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/331,002	06/14/1999	MICHAEL TASLER	2055/101	1117

TITLE OF INVENTION: FLEXIBLE INTERFACE FOR COMMUNICATION BETWEEN A HOST AND AN ANALOG I/O DEVICE CONNTECTED TO THE INTERFACE REGARDLESS THE TYPE OF THE I/O DEVICE

APPLN. TYPE	SMALL ENTITY	ISSUE FEE	PUBLICATION FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	YES	$640	$0	$640	08/19/2002

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE REFLECTS A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE APPLIED IN THIS APPLICATION. THE PTOL-85B (OR AN EQUIVALENT) MUST BE RETURNED WITHIN THIS PERIOD EVEN IF NO FEE IS DUE OR THE APPLICATION WILL BE REGARDED AS ABANDONED.

HOW TO REPLY TO THIS NOTICE:

I. Review the SMALL ENTITY status shown above. If the SMALL ENTITY is shown as YES, verify your current SMALL ENTITY status:

A. If the status is changed, pay the PUBLICATION FEE (if required) and twice the amount of the ISSUE FEE shown above and notify the United States Patent and Trademark Office of the change in status, or

B. If the status is the same, pay the TOTAL FEE(S) DUE shown above.

If the SMALL ENTITY is shown as NO:

A. Pay TOTAL FEE(S) DUE shown above, or

B. If applicant claimed SMALL ENTITY status before, or is now claiming SMALL ENTITY status, check the box below and enclose the PUBLICATION FEE and 1/2 the ISSUE FEE shown above.

☐ Applicant claims SMALL ENTITY status. See 37 CFR 1.27.

II. PART B - FEE(S) TRANSMITTAL should be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). Even if the fee(s) have already been paid, Part B - Fee(s) Transmittal should be completed and returned. If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Box ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PTOL-85 (REV. 04-02) Approved for use through 01/31/2004.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail Box ISSUE FEE
Commissioner for Patents
Washington, D.C. 20231
Fax (703)746-4000

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 4 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Legibly mark-up with any corrections or use Block 1)
24283 7590 05/17/2002

PATTON BOGGS
PO BOX 270930
LOUISVILLE, CO 80027

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Box Issue Fee address above, or being facsimile transmitted to the USPTO, on the date indicated below.

(Depositor's name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/331,002	06/14/1999	MICHAEL TASLER	2055/101	1117

TITLE OF INVENTION: FLEXIBLE INTERFACE FOR COMMUNICATION BETWEEN A HOST AND AN ANALOG I/O DEVICE CONNTECTED TO THE INTERFACE REGARDLESS THE TYPE OF THE I/O DEVICE

APPLN. TYPE	SMALL ENTITY	ISSUE FEE	PUBLICATION FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	YES	$640	$0	$640	08/19/2002

EXAMINER	ART UNIT	CLASS-SUBCLASS
DU, THUAN N	2185	710-001000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

❑ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.

❑ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list (1) the names of up to 3 registered patent attorneys or agents OR, alternatively, (2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1 _____
2 _____
3 _____

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. Inclusion of assignee data is only appropriate when an assignment has been previously submitted to the USPTO or is being submitted under separate cover. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE (B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) ❑ individual ❑ corporation or other private group entity ❑ government

4a. The following fee(s) are enclosed:

❑ Issue Fee
❑ Publication Fee
❑ Advance Order - # of Copies _____

4b. Payment of Fee(s):

❑ A check in the amount of the fee(s) is enclosed.
❑ Payment by credit card. Form PTO-2038 is attached.
❑ The Commissioner is hereby authorized by charge the required fee(s), or credit any overpayment, to Deposit Account Number _____(enclose an extra copy of this form).

Commissioner for Patents is requested to apply the Issue Fee and Publication Fee (if any) or to re-apply any previously paid issue fee to the application identified above.

(Authorized Signature) (Date)

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, Washington, D.C. 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, Washington, DC 20231.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMIT THIS FORM WITH FEE(S)

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/331,002	06/14/1999	MICHAEL TASLER	2055/101	1117

24283 7590 05/17/2002

PATTON BOGGS
PO BOX 270930
LOUISVILLE, CO 80027
UNITED STATES

EXAMINER	
DU, THUAN N	
ART UNIT	**PAPER NUMBER**
2185	

DATE MAILED: 05/17/2002

Determination of Patent Term Extension under 35 U.S.C. 154 (b)
(application filed after June 7, 1995 but prior to May 29, 2000)

The patent term extension is 0 days. Any patent to issue from the above identified application will include an indication of the 0 day extension on the front page.

If a continued prosecution application (CPA) was filed in the above-identified application, the filing date that determines patent term extension is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) system. (http://pair.uspto.gov)

PTOL-85 (REV. 04-02) Approved for use through 01/31/2004.

Exhibit M

Reproduced By GLOBAL ENGINEERING DOCUMENTS
With The Permission Of ANSI
Under Royalty Agreement

American National Standard

for Information Systems –

Small Computer System Interface-2

ANSI American National Standards Institute
11 West 42nd Street
New York, New York
10036

ANSI X3.131-1994

ANSI®
X3.131-1994

American National Standard
for Information Systems –

Small Computer System Interface-2

Secretariat

Computer and Business Equipment Manufacturers Association

Approved January 31, 1994

American National Standards Institute, Inc.

Abstract

The SCSI protocol is designed to provide an efficient peer-to-peer I/O bus with up to 16 devices, including one or more hosts. Data may be transferred asynchronously at rates that only depend on device implementation and cable length. Synchronous data transfers are supported at rates up to 10 mega-transfers per second. With the 32-bit wide data transfer option, data rates of up to 40 megabytes per second are possible.

SCSI-2 includes command sets for magnetic and optical disks, tapes, printers, processors, CD-ROMs, scanners, medium changers, and communications devices.

American National Standard

Approval of an American National Standard requires verification by ANSI that the requirements for due process, consensus, and other criteria for approval have been met by the standards developer.

Consensus is established when, in the judgment of the ANSI Board of Standards Review, substantial agreement has been reached by directly and materially affected interests. Substantial agreement means much more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that a concerted effort be made toward their resolution.

The use of American National Standards is completely voluntary; their existence does not in any respect preclude anyone, whether he has approved the standards or not, from manufacturing, marketing, purchasing, or using products, processes, or procedures not conforming to the standards.

The American National Standards Institute does not develop standards and will in no circumstances give an interpretation of any American National Standard. Moreover, no person shall have the right or authority to issue an interpretation of an American National Standard in the name of the American National Standards Institute. Requests for interpretations should be addressed to the secretariat or sponsor whose name appears on the title page of this standard.

CAUTION NOTICE: This American National Standard may be revised or withdrawn at any time. The procedures of the American National Standards Institute require that action be taken periodically to reaffirm, revise, or withdraw this standard. Purchasers of American National Standards may receive current information on all standards by calling or writing the American National Standards Institute.

CAUTION: The developers of this standard have requested that holders of patents that may be required for the implementation of the standard disclose such patents to the publisher. However, neither the developers nor the publisher have undertaken a patent search in order to identify which, if any, patents may apply to this standard.

As of the date of publication of this standard and following calls for the identification of patents that may be required for the implementation of the standard, no such claims have been made. No further patent search is conducted by the developer or publisher in respect to any standard it processes. No representation is made or implied that licenses are not required to avoid infringement in the use of this standard.

Published by

American National Standards Institute
11 West 42nd Street, New York, New York 10036

Copyright ©1994 by American National Standards Institute
All rights reserved.

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

Printed in the United States of America

APS1M894/125

CONDITION status with the sense key set to MISCOMPARE. The remaining fields in the sense data shall be set as documented in the COPY command.

8.2.5 INQUIRY command

The INQUIRY command (see table 44) requests that information regarding parameters of the target and its attached peripheral device(s) be sent to the initiator. An option allows the initiator to request additional information about the target or logical unit (see 8.2.5.2).

Table 44 - INQUIRY command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (12h)							
1	Logical unit number			Reserved				EVPD
2	Page code							
3	Reserved							
4	Allocation length							
5	Control							

An enable vital product data (EVPD) bit of one specifies that the target shall return the optional vital product data specified by the page code field. If the target does not support vital product data and this bit is set to one, the target shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB.

An EVPD bit of zero specifies that the target shall return the standard INQUIRY data. If the page code field is not zero, the target shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB.

The page code field specifies which page of vital product data information the target shall return (see 8.3.4).

The INQUIRY command shall return CHECK CONDITION status only when the target cannot return the requested INQUIRY data.

NOTE 64 The INQUIRY data should be returned even though the peripheral device may not be ready for other commands.

If an INQUIRY command is received from an initiator with a pending unit attention condition (i.e. before the target reports CHECK CONDITION status), the target shall perform the INQUIRY command and shall not clear the unit attention condition (see 7.9).

NOTES
65 The INQUIRY command is typically used by the initiator after a reset or power-up condition to determine the device types for system configuration. To minimize delays after a reset or power-up condition, the standard INQUIRY data should be available without incurring any media access delays. If the target does store some of the INQUIRY data on the device, it may return zeros or ASCII spaces (20h) in those fields until the data is available from the device.
66 The INQUIRY data may change as the target executes its initialization sequence or in response to a CHANGE DEFINITION command. For example, the target may contain a minimum command set in its non-volatile memory and may load its final firmware from the device when it becomes ready. After it has loaded the firmware, it may support more options and therefore return different supported options information in the INQUIRY data.

8.2.5.1 Standard INQUIRY data

The standard INQUIRY data (see table 45) contains 36 required bytes, followed by a variable number of vendor-specific parameters. Bytes 56 through 95, if returned, are reserved for future standardization.

Table 45 - Standard INQUIRY data format

Bit / Byte	7	6	5	4	3	2	1	0
0	Peripheral qualifier			Peripheral device type				
1	RMB	Device-type modifier						
2	ISO version		ECMA version			ANSI-approved version		
3	AENC	TrmIOP	Reserved		Response data format			
4	Additional length (n-4)							
5	Reserved							
6	Reserved							
7	RelAdr	WBus32	WBus16	Sync	Linked	Reserved	CmdQue	SftRe
8 – 15	(MSB) Vendor identification (LSB)							
16 – 31	(MSB) Product identification (LSB)							
32 – 35	(MSB) Product revision level (LSB)							
36 – 55	Vendor-specific							
56 – 95	Reserved							
	Vendor-specific parameters							
96 – n	Vendor-specific							

The peripheral qualifier and peripheral device-type fields identify the device currently connected to the logical unit. If the target is not capable of supporting a device on this logical unit, this field shall be set to 7Fh (peripheral qualifier set to 011b and peripheral device type set to 1Fh). The peripheral qualifier is defined in table 46 and the peripheral device type is defined in table 47.

Table 46 - Peripheral qualifier

Qualifier	Description
000b	The specified peripheral device type is currently connected to this logical unit. If the target cannot determine whether or not a physical device is currently connected, it shall also use this peripheral qualifier when returning the INQUIRY data. This peripheral qualifier does not mean that the device is ready for access by the initiator.
001b	The target is capable of supporting the specified peripheral device type on this logical unit; however, the physical device is not currently connected to this logical unit.
010b	Reserved
011b	The target is not capable of supporting a physical device on this logical unit. For this peripheral qualifier the peripheral device type shall be set to 1Fh to provide compatibility with previous versions of SCSI. All other peripheral device type values are reserved for this peripheral qualifier.
1XXb	Vendor-specific

Table 47 - Peripheral device type

Code	Description
00h	Direct-access device (e.g. magnetic disk)
01h	Sequential-access device (e.g. magnetic tape)
02h	Printer device
03h	Processor device
04h	Write-once device (e.g. some optical disks)
05h	CD-ROM device
06h	Scanner device
07h	Optical memory device (e.g. some optical disks)
08h	Medium changer device (e.g. jukeboxes)
09h	Communications device
0Ah - 0Bh	Defined by ASC IT8 (Graphic arts pre-press devices)
0Ch - 1Eh	Reserved
1Fh	Unknown or no device type

A removable medium (RMB) bit of zero indicates that the medium is not removable. A RMB bit of one indicates that the medium is removable.

The device-type modifier field was defined in SCSI-1 to permit vendor-specific qualification codes of the device type. This field is retained for compatibility with SCSI-1. Targets that do not support this field should return a value of zero.

The usage of non-zero code values in the ISO version and ECMA version fields are defined by the International Organization for Standardization and the European Computer Manufacturers Association, respectively. A zero code value in these fields shall indicate that the target does not claim compliance to the ISO version of SCSI (ISO 9316) or the ECMA version of SCSI (ECMA-111). It is possible to claim compliance to more than one of these SCSI standards.

The ANSI-approved version field indicates the implemented version of this standard and is defined in table 48.

Table 48 - ANSI-approved version

Code	Description
0h	The device might or might not comply to an ANSI-approved standard.
1h	The device complies to ANSI X3.131-1986 (SCSI-1).
2h	The device complies to this version of SCSI. This code is reserved to designate this standard upon approval by ANSI.
3h - 7h	Reserved

The asynchronous event notification capability (AENC) bit indicates that the device supports the asynchronous event notification capability as defined in 7.5.5.

a) Processor device-type definition: An AENC bit of one indicates that the processor device is capable of accepting asynchronous event notifications. An AENC bit of zero indicates that the processor device does not support asynchronous event notifications.
b) All other device-types: This bit is reserved.

A terminate I/O process (TrmIOP) bit of one indicates that the device supports the TERMINATE I/O PROCESS message as defined in 6.6.22. A value of zero indicates that the device does not support the TERMINATE I/O PROCESS message.

A response data format value of zero indicates the INQUIRY data format is as specified in SCSI-1. A response data format value of one indicates compatibility with some products that were designed prior to the development of this standard (i.e. CCS). A response data format value of two indicates that the data shall be in the format specified in this standard. Response data format values greater than two are reserved.

The additional length field shall specify the length in bytes of the parameters. If the allocation length of the command descriptor block is too small to transfer all of the parameters, the additional length shall not be adjusted to reflect the truncation.

A relative addressing (RelAdr) bit of one indicates that the device supports the relative addressing mode for this logical unit. If this bit is set to one, the linked command (Linked) bit shall also be set to one; since relative addressing can only be used with linked commands. A RelAdr bit of zero indicates the device does not support relative addressing for this logical unit.

A wide bus 32 (Wbus32) bit of one indicates that the device supports 32-bit wide data transfers. A value of zero indicates that the device does not support 32-bit wide data transfers.

A wide bus 16 (Wbus16) bit of one indicates that the device supports 16-bit wide data transfers. A value of zero indicates that the device does not support 16-bit wide data transfers.

NOTE 67 If the values of both the Wbus16 and Wbus32 bits are zero, the device only supports 8-bit wide data transfers.

A synchronous transfer (Sync) bit of one indicates that the device supports synchronous data transfer. A value of zero indicates the device does not support synchronous data transfer.

A linked command (Linked) bit of one indicates that the device supports linked commands for this logical unit. A value of zero indicates the device does not support linked commands for this logical unit.

A command queuing (CmdQue) bit of one indicates that the device supports tagged command queuing for this logical unit. A value of zero indicates the device does not support tagged command queuing for this logical unit.

A soft reset (SftRe) bit of zero indicates that the device responds to the RESET condition with the hard RESET alternative (see 6.2.2.1). A SftRe bit of one indicates that the device responds to the RESET condition with the soft RESET alternative (see 6.2.2.2).

ASCII data fields shall contain only graphic codes (i.e. code values 20h through 7Eh). Left-aligned fields shall place any unused bytes at the end of the field (highest offset) and the unused bytes shall be filled with space characters (20h). Right-aligned fields shall place any unused bytes at the start of the field (lowest offset) and the unused bytes shall be filled with space characters (20h).

The vendor identification field contains eight bytes of ASCII data identifying the vendor of the product. The data shall be left aligned within this field.

NOTE 68 It is intended that this field provide a unique vendor identification of the manufacturer of the SCSI device. In the absence of a formal registration procedure, X3T10 maintains a list of vendor identification codes in use. Vendors are requested to voluntarily submit their identification codes to X3T10 to prevent duplication of codes (see annex E).

The product identification field contains sixteen bytes of ASCII data as defined by the vendor. The data shall be left-aligned within this field.

The product revision level field contains four bytes of ASCII data as defined by the vendor. The data shall be left-aligned within this field.

8.2.5.2 Vital product data

Implementation of vital product data is optional. The information returned consists of configuration data (e.g. vendor identification, product identification, model, serial number), manufacturing data (e.g. plant and date of manufacture), field replaceable unit data and other vendor- or device-specific data.

The initiator requests the vital product data information by setting the EVPD bit to one and specifying the page code of the desired vital product data (see 8.3.4). If the target does not implement the requested page it shall return CHECK CONDITION status. The a sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN CDB.

NOTES
69 The target should have the ability to execute the INQUIRY command even when a device error occurs that prohibits normal command execution. In such a case, CHECK CONDITION status should be returned for commands other than INQUIRY or REQUEST SENSE. The sense data returned may contain the field replaceable unit code. The vital product data should be obtained for the failing device using the INQUIRY command.
70 This standard defines a format that allows device-independent initiator software to display the vital product data returned by the INQUIRY command. For example, the initiator may display the data associated for the field replaceable unit returned in the sense data. The contents of the data may be vendor-specific; therefore, it may not be usable without detailed information about the device.
71 This standard does not define the location or method of storing the vital product data. The retrieval of the data may require completion of initialization operations within the device that may induce delays before the data is available to the initiator. Time-critical requirements are an implementation consideration and are not addressed in this standard.

Exhibit N

Copyright
by
Michael Tasler
1996

Design and Construction of a Universal Data Acquisition and Control System for Scanning Probe Microscopy

by

Michael Tasler

THESIS

Presented to the Faculty of the Graduate School
of The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF ARTS

THE UNIVERSITY OF TEXAS AT AUSTIN

May, 1996

Design and Construction of a Universal Data Acquisition and Control System for Scanning Probe Microscopy

APPROVED BY

SUPERVISING COMMITTEE:

Chih-Kang Shih

John Markert

THIS IS AN ORIGINAL MANUSCRII
IT MAY NOT BE COPIED WITHOUT
THE AUTHOR'S PERMISSION

ACKNOWLEDGMENTS

I would like to first acknowledge my advisor Prof. Ken Shih. Thanks to his financial support it was possible for me to finish my studies at the University of Texas at Austin.

I greatly appreciate the tremendous help I received from Prof. Max Scheer and Prof. Manfred Fink, both of whom invested their time in the German partnership program.

I also would like to thank Prof. Keto for his insightful suggestions on various topics.

I also received important support from Texas Instruments Inc., Analog Devices and Motorola, who sponsored my work with sample ICs.

Last but definitely not least I want to acknowledge the enormous help I received during countless conversations with my friends in Dr. Shih's lab and Dr. Martinez' lab. Especially Günter Guttroff who spent quite some time listening to my plans. Gordon Chao and Art Smith were always there to answer my questions and introduce me to the sub-nanometer world.

Ricardo Claps, Andrew McDonald, Michael Barrett and Doris Heinrich turned out to be the source of new motivation to keep me working on my project. I greatly appreciate the time we talked during the breaks we took together.

ABSTRACT

Design and Construction of a Universal Data Acquisition and Control System for Scanning Probe Microscopy

by

Michael Tasler, M.A.

The University of Texas at Austin, 1996

SUPERVISOR: Chih-Kang Shih

Scanning Probe Microscopy (SPM) has been used to study many novel material properties such as single electronic energy levels formed in a quantum corral system. However, electronic states of more realistic structures like quantum wells, which are of interest in optoelectronic applications, have not yet been fully explored using SPM. While this is possible in principle, it requires a more sophisticated SPM control unit which is not available commercially. The development of such a system is the aim of this project. General ideas and problems are discussed.

TABLE OF CONTENTS

ACKNOWLEDGMENTS ..... **iv**

ABSTRACT ..... **v**

Chapter 1: Introduction **1**

(1.1) Motivation: 1

(1.2) General SPM Setup: 2

(1.2.1) General Mechanical SPM Design: 2

(1.2.2) Electrical SPM Setup in the Case of Tunneling Microscopy: 4

Chapter 2: Design of the SPM Control Unit **8**

(2.1) Theoretical Discussion of the System Requirements and Limits: 8

(2.1.1) Electronic Noise: 8

(2.1.1.1) Shot Noise: 8

(2.1.1.2) Johnson Noise: 9

(2.1.1.3) Flicker Noise: 9

(2.1.1.4) Noise of a JFET Preamplifier: 9

(2.1.2) Required accuracy: 10

(2.1.2.1) Accuracy of the Tip Positioning: 10

(2.1.2.2) Required Accuracy of the Bias Voltage: 11

(2.2) Practical Realizations: 12

(2.2.1) Timing Requirements for Different Scanning Modes: 13

(2.2.2) The Digital Control Unit: 14

(2.3) The Realization of an Independent Processor System: 14

(2.4) Concluding Comments on System Limits: 16

Chapter 3: Hardware Design **17**
(3.1) The DSP Motherboard: 18
(3.1.1) The Interrupt Manager (IM): 18
(3.1.2) The Connection to the Master PC: 19
(3.2) The TLC320AC01 Serial Port: 23
(3.4) The Parallel Digital I/O Port: 24
(3.5) The Analog Boards: 24

Chapter 4: The Final Assembly **26**

Chapter 5: The Software **29**
(5.1) TI's Assembler: 29
(5.2) The DSK Programmer Tool: 30
(5.3) Dynamic Range Expansion: 33

Chapter 6: Results and Conclusions **37**

APPENDIX **38**
Appendix A: Block Diagram of SPM Control Unit 39
Appendix B: Schematics 40
Appendix C: PCB Layouts 43
Appendix D: The PC Software 57
Appendix E: The DSP Software 68

BIBLIOGRAPHY **86**
VITA **87**

Chapter 1: Introduction

(1.1) Motivation:

The interest in Scanning Probe Microscopy (SPM) has been increasing ever since the Scanning Tunneling Microscopy (STM) was invented by G. Binnig and H. Rohrer in 1981. Several novel technologies were developed in the following years to investigate many different surface properties. All of those techniques share the task of discovering, exploring and resolving more precise information of an expanding variety of surface structures and materials. The decreasing size of those structures give rise to new effects like the unique electronic properties of quantum heterostructures. As a consequence the design and the control of the scanning probe microscopes and data acquisition has to fulfill higher standards.

While the recent quality of the mechanical design of scanning probe microscopes already allows theoretically scanning as precise as about 1/500 Å, many control and data acquisition systems are less accurate. However, the question to be solved is where the physically meaningful limit of scanning control is to be set. Temperature related effects can only be minimized within a certain range by changing the environmental temperature of the probe and the microscope. Time dependent disturbances of the signal stability, like drift effects, influence the method of measuring as well as disturbances caused by the observation technique itself. Thus a very thorough discussion of system requirements and inevitable disturbances is essential. Only then a reasonable optimization of the control and data acquisition units is practicable.

To achieve the determination and the spatial resolution of electronic states within a GaAs/AlGaAs/GaAs quantum well formation with a STM is my future goal and therefore the overall motivation, which is used as a guideline throughout the work. At the same time it is

to be kept in mind to maintain a high flexibility of the system in order to fulfill the needs of other SPM experiments.

(1.2) General SPM Setup:

The general idea of all SPMs is to place a sharp tip close enough to the surface of interest so that atomic interactions between tip and sample can occur. The magnitude of the detected interaction signal is dependent on the effective separation of tip and sample. To keep this signal constant is the task of a feedback circuit, which can adjust the tip to sample distance for that purpose. If this feedback system is turned on and two dimensional scanning across the surface is started, the tip position will reflect a plane of constant interaction of tip and sample.

(1.2.1) General Mechanical SPM Design:

Three dimensional nanoscale tip positioning is commonly done with piezoelectric devices, which expand and contract roughly proportional to an applied voltage. One possible design is a scanning tube as shown in figure 1.1.

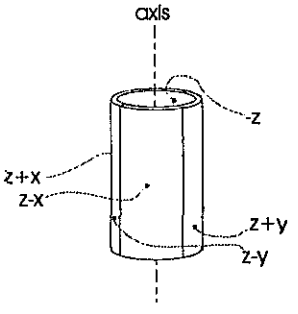


Fig 1.1: Scanning tube. The drawing shows the four segments of the tube and their wiring including the common center contact.

If different voltages are applied on opposite sections of the tube, their different expansions will lead to a bending of the tube and therefore to a movement of the axial mounted tip. This movement is roughly perpendicular to the axis. By changing the offset voltage of all segments simultaneously, the tube length will change steadily and move the tip in axial direction. Positioning within several μm range is achievable with this technique.

Although this range is plenty for atomic resolution scanning, it is too small for practical purposes. Thus a micropositioning unit is necessary, to bring the sample into the scan range of the tip. A widely used technique is a two dimensional translation stage (Walker) as shown in figure 1.2.

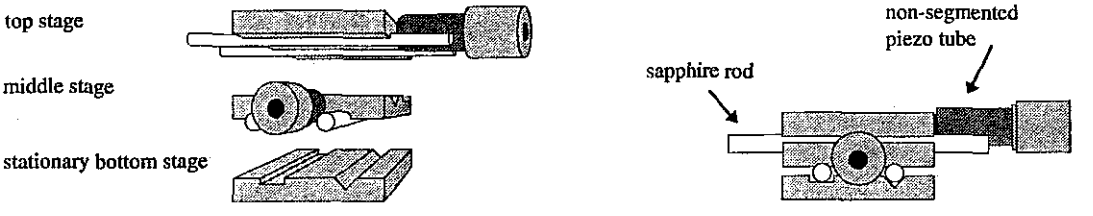


Fig 1.2: Walker design

The basic idea is to provide two perpendicular orientated motion directions. Each is represented by an independent stage of the walker. The driving force is generated by nonsegmented piezo tubes which are mounted with one end to the stage and with the other to a counterweight. If a voltage is applied so that the length of the piezo is changed, a force will act on the stage and the counterweight. Due to friction of the stage, the force has to exceed a certain threshold before motion will occur. The wave form of the applied voltage signal thus determines direction and speed of the movement. Optimal movement is achieved by cycloids as they are shown in figure 1.3. Each cycle lets the walker jump by about 50 nm. The

average step size can be adjusted by varying the frequency and the amplitude of the applied signals. Common settings are 225 V peak voltage and 1 kHz cycle frequency.

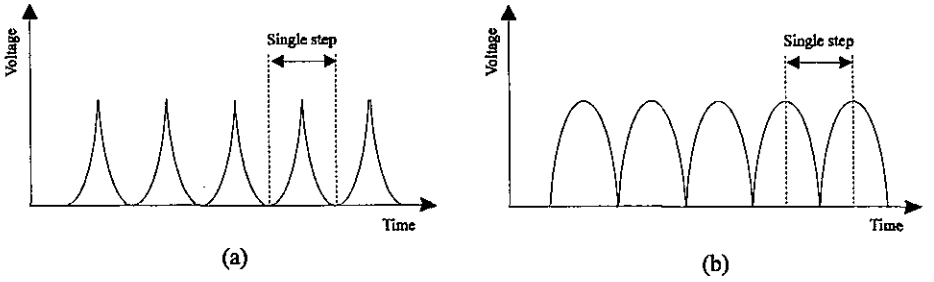


Fig. 1.3: Driving signal for walker piezo tube for (a) forward and (b) backward movement

The function for the time dependence of the voltage in figure (1.3) is:

$$(a)\quad V(t) = V_0 \cdot \sqrt{1-x^2}\ ,\ |x| \le 1$$

$$(b)\quad V(t) = V_0 \cdot \left(1-\sqrt{1-x^2}\right),\ |x| \le 1$$

where V_0 is the maximum voltage.

(1.2.2) Electrical SPM setup in the case of Tunneling Microscopy:

According to the above mentioned overall motivation of doing a certain STM study, the general description of SPMs is replaced in this chapter by a more detailed introduction into the electrical schematic of a Scanning Tunneling Microscope.

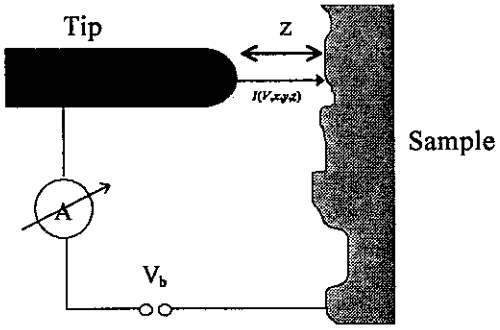


Fig. 1.4: Simplified plot of the STM constellation

If the tip to sample distance z is small enough, a tunneling current $\mathbf{I_t}$ is detected. This current is dependent of the sign and the absolute value of the applied bias voltage $\mathbf{V_b}$ and reflects directly the Local Density of States Contour. It decays exponentially with the distance **z**:

$$I \sim \exp(-\kappa z)$$

where

$$\kappa = \sqrt{\frac{2m\varepsilon}{\hbar^2}} \approx \frac{1}{\text{Å}}$$

and ε is the energy shown in the figure. Here, $\varepsilon_{f,t}$ and $\varepsilon_{f,s}$ are the Fermi levels of the tip and the sample. The required energy to move an electron away from either the tip or the sample is described by ϕ_t and ϕ_s which are the work functions of tip and sample.

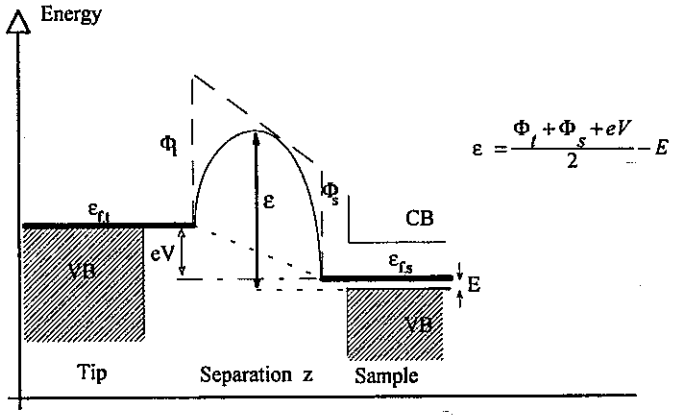


Fig. 1.4: Energy diagram of the tunneling junction. (VB) is the Valence band, (CB) is the Conduction band. E is the energy of the electronic state of the sample, which is involved in the tunneling process.

The tunneling current at a given voltage V and position z can be expressed by:

$$I(V) = \int \rho_s(E) \cdot \rho_t(E - eV) \cdot \exp[-2\kappa(E) z] dE$$

where ρ_t and ρ_s are the local density of electronic states of the tip and the sample.

As a result the direct mapping of the energy distribution of the electronic local density of states is made possible by detecting the tunneling current $\mathbf{I_t}$ at a given position **(x, y)** and distance **z**, while the applied bias voltage $\mathbf{V_b}$ is scanned over a given range. Due to the exponential decay of $\mathbf{I_t}$ with **z**, a precise and stable tip position is very essential. However, this z dependence can be used to expand the dynamic resolution of the acquisition system. By moving the tip closer to the sample the current signal can be magnified by several orders if necessary. This technique is of interest if the bias voltage is scanned over a wide range including zero voltage.

However, the complex interaction of **z**, $\mathbf{V_b}$ and $\mathbf{I_t}$ must be studied well before any result can be derived from the set of data obtained by this method of **z** and $\mathbf{V_b}$ variation. This was first done by Feenstra *et al.* [1], who established the simultaneously scanning of **z** and $\mathbf{V_b}$ where

$$z = \xi V_b + z_{\min}.$$

Although this method allows a better insight in the energy distribution of electronic states it still bears several uncertainties. Physically relevant results can only be derived if all data values refer to the same reference position $\mathbf{z_0}$, which turns out to be a major problem of the simultaneous **V** and **z** ramping method, since no pair of two adjacent values can be found, where either z or V is identical.

How to choose a relation, which estimates the **z** and **V** dependence of $\mathbf{I_t}$ plays a very essential role in relating all data values back to a constant reference position $\mathbf{z_0}$ and will therefore affect the derived physical results of the measurement.

It is part of this work to introduce a new method to increase the dynamic resolution of $\mathbf{I_t}$ with a different **z** variation technique.

Chapter 2: Design of the SPM Control Unit

(2.1) Theoretical Discussion of the System Requirements and Limits:

As mentioned in the introduction, the discussion of system requirements is basically done with respect to a STM study of the electronic states of an GaAs/AlGaAs/GaAs quantum well.

(2.1.1) Electronic Noise:

Recently electronic noise of the STM controlling system and its non-expandable design are limiting the capabilities of the STM. How much the intrinsic, unavoidable noises, like shot noise, Johnson noise and flicker noise, are limiting the expectable resolution is discussed in this chapter as well as thermal drift effects and filtering.

(2.1.1.1) Shot Noise:

Statistic fluctuations of any current I, which are caused by the finite quantization of the electric charges, lead to an inevitable white noise. This fluctuation has a Gaussian amplitude distribution and is known as shot noise. Since this noise is white and the relative fluctuation is larger for small currents, good filtering and a narrow bandwidth Δf is essential for the precise detection of the low tunneling current.

Assuming that 40,000 data points are taken each second, Δf is limited to 40 kHz by the relation $\Delta f \cdot \Delta t \geq 1$. For a tunneling current of 1 nA, the rms fluctuation is:

$$I_{shot\ noise}(rms) = \sqrt{2eI_t\Delta f} \approx \sqrt{2 \cdot 1.6 \cdot 10^{-19}\,\mathrm{As} \cdot 1\ \mathrm{nA} \cdot 40\ \mathrm{kHz}} = 3.6\ \mathrm{pA} = 0.36\% \cdot \mathrm{I_t}\ .$$

(2.1.1.2) Johnson noise:

In order to be able to analyze the tunneling current, a conversion to voltage and a preamplification of the signal within the UHV chamber of the STM by a factor of 10^7 is done. The basic idea is to measure the voltage, which falls off at a 10 MΩ resistor, while $\mathbf{I_t}$ is flowing through it. Additional to the shot noise, a temperature dependent white noise occurs, which is created by fluctuations of this resistance. The rms fluctuation of this so called Johnson noise is like the shot noise dependent on the bandwidth of the acquisition system. Assuming that the temperature **T** of the chamber is 80 K and that Δf is limited to 40 kHz, the effect of this noise is:

$$\frac{V_{Johnson\ noise}(rms)}{R I_t} = \frac{\sqrt{4 k_B T R \Delta f}}{R I_t} = 0.42\%$$

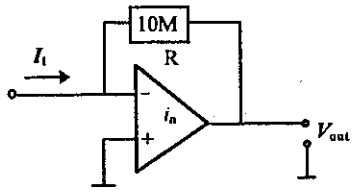
(2.1.1.3) Flicker noise:

A third kind of noise is the flicker noise, generated by the resistor as well. It has typically a 1/f dependence and causes a maximum rms noise of 0.2 μV per decade for metal-film resistors. Assuming again a bandwidth of Δf = 40 kHz, which is 4.5 decades, this noise will not exceed the 1 μV limit (which is less than 0.01% of the signal voltage).

(2.1.1.4) Noise of a JFET Preamplifier:

To obtain a low impedance output, a JFET instrumentation amplifier is used. The overall error, caused by the previously mentioned effects can be estimated for the amplifying system

according to Horowitz [3]. The internal noise $i_n/\sqrt{f}$ of JFETs below 50 kHz is about 5 fA/Hz^{½}.



The noise of this inverting unit can be expressed by the relation:

$$\frac{\Delta I}{I} = \frac{\sqrt{\left(i_n^2 + 4k_B T \frac{1}{R}\right) \cdot \Delta f}}{I} \approx 0{,}4\% \quad \text{for } I = 1 \text{ nA}.$$

The accuracy of the whole system is therefore limited to 0.4%.

(2.1.2) Required accuracy:

In order to get a rough estimate for the accuracy required of the tip positioning and the control signals, we need to compute their effect on the tunneling current and to claim that those influences do not exceed the 0.4% level of the intrinsic noise.

(2.1.2.1) Accuracy of the Tip Positioning:

First of all, the tip vibration Δz must be considered. According to the approximation

$$I - \Delta I \propto e^{-\kappa(z+\Delta z)} = e^{-\kappa z} e^{-\kappa \Delta z} \approx e^{-\kappa z}(1-\kappa \Delta z)$$
$$\Rightarrow \frac{I-\Delta I}{I} = 1-\kappa \Delta z \;\Rightarrow \frac{\Delta I}{I} = \kappa \Delta z \overset{!}{\leq} 0.4\%$$

and with a realistic decay constant of $\kappa = 1.1 \cdot 10^{10}\, m^{-1}$, Δz must fulfill:

$\Delta z \leq \frac{1}{250}$ Å. That is mechanically already practicable.

(2.1.2.2) Required Accuracy of the Bias Voltage:

Another point is the ramping of the bias voltage. Referring to Feenstra's research [2], the tunneling current shows a proportionality to $(V\text{-}V_0)^{\alpha}$, where eV_0 is the band edge energy and $\alpha \approx 1.5$. From this information, it is possible to derive:

$$\frac{1}{I}\frac{dI}{dV} \approx \frac{\alpha}{V-V_0} \Rightarrow \frac{dI}{I} \approx \alpha \frac{dV}{V-V_0} \overset{!}{\leq} 0.4\%,$$

$$\Rightarrow \; (V-V_0) \geq 370 \cdot \Delta V_{step}.$$

This determines the minimal value of the energy resolution around a band edge to be three to four hundred times larger than the stepping size of the voltage ramp.

At the given temperature $T = 80$ K, which means that $k_B T \approx 7$ meV, it does not make sense to look for any details below 10 meV resolution. If we want to resolve details at this limit, which means that $(V\text{-}V_0)$ is minimal 10 mV, the corresponding maximal ΔV step size can be estimated by using the above derived relation to be less than 30 μV. In the case that we are scanning over a ±2.5 V bias voltage range with ten ascending intervals of 0.5 V length, every window requires a resolution of

$\frac{30\mu V}{0.5V} = \frac{1}{1.7 \cdot 10^4}$, which equals exactly 14 bit accuracy of AD and DA ICs. To obtain this resolution over the entire ±2.5 V range, a 17 or 18 bit AD/DA system would be required. That is far beyond a trivial task and exceeds my abilities. Therefore I concentrated on smaller windows of 0.5 V range each.

(2.2) Practical Realizations:

Besides the unavoidable noises, other influences like 60 Hz and 120 Hz line noise and thermal drift effects are reducing the stability and the resolution of the system.

It seems therefore to be very essential to keep the length of all signal wires short. One idea is to place the AD and DA boards within the UHV chamber of the SPM itself. First of all, the data acquisition would be closer to the origin of the signals. Secondly the metal surface of the chamber would act like a shielding system and the lower temperature in the chamber would reduce the above mentioned intrinsic noise. The only problems are that the electronic devices are not running at temperatures as low as 80K and that the number of connections into the UHV chamber is limited. The circuit boards must therefore be located in a different part of the chamber than the sample itself, which might cause some difficulties. The number of wires is kept low by providing fast serial ports which are optimized for different purposes. To explain those purposes it is helpful to think about various modes of the data acquisition, especially concerning the timing of the in- and outgoing signals.

(2.2.1) Timing Requirements for different Scanning Modes:

There are basically two different ways of using an STM: the constant current topography (CCT) mode and the constant current tunneling spectroscopy (CITS) mode.
In the CCT mode the tip scans continuously in **x** and **y** direction over the sample. The feedback circuit has to adjust **z** to provide a constant tunneling current. The measured information is **z**.
For this purpose a smooth **x** ramp and a stable, uniformly stepping **y** signal are needed. Good low pass filtering and good timing can increase the quality of the DA output signals. The feedback circuit should respond quickly and almost unaffected by any kind of filtering. In order to realize a high flexibility of the system, a digital feedback should be provided. Thus the adjustment of any feedback parameters can be easily done by software.
Those tasks make two basic sets of data converters necessary:
a medium fast set of ADCs and DACs with programmable timers, filters and gain
and a fast set of DACs with a high output slew rate.

In addition to this scanning technique an energy mapping can be done at each x,y position to obtain information about electronic states of the sample surface. Since the energy relates to the bias voltage, a measurement of $I_t(V)$ provides a method of mapping the energy distribution of the density of electronic states. This mode is called CITS. It requires stable and uniformly stepping **x** and **y** signals to scan in single steps over the sample. After each step the feedback is turned off and both **V** and **z** are scanned simultaneously and smoothly over the given range while $I_{x,y}(z,V)$ is measured. This requires slow **x** and **y** outputs, medium-fast, simultaneously timed **V** and **z** outputs, and a fast feedback response output.

Beside nontrivial programming this comes down to the same two kinds of needed sets of hardware as for the first discussed CCT mode. The slow x and y signals can be generated by medium fast DACs with a very low filtering frequency.

(2.2.2) The Digital Control Unit:

Not yet mentioned is the load of information which must be handled by the master unit of the acquisition system, which in our case is an IBM compatible PC. Although the processor of a modern PC might be fast enough to handle this amount of data, it is impossible to get it all in and out in time through any standard PC port. Thus a secondary device is to be designed, that takes over the complete SPM control, but that itself is driven by a master PC. In this case only the starting conditions and the final results must be send and received by the PC.

Especially for the fast analysis and control of multiple signal systems DSPs, Digital Signal Processors, are available. After comparing the features of many different types of DSPs, I finally chose the TMS320C50 from Texas Instruments. The advantage of this DSP is that it has a relatively large internal memory, a big set of highly optimized commands, fast and easy programmable bus systems, and the fact that a starter kit with very good manuals is offered. This starter kit provides all the features of the single DSP chip, but already contains boot software, which allows the convenient programming over the COM port of the PC.

(2.3) The Realization of an independent Processor System:

To allow a multiple timer system to work properly, a sophisticated way of handling the requests for processor time must be found. A well known way are interrupt-requests, IRQs, which interrupt the processor immediately to start to work at another part of the program. The problem is that some processes are so important that they cannot be interrupted and others are so important that they have to interrupt immediately. Therefore an interrupt manager (IM) with freely definable priorities is desired. It is very important that this IM

must not have any dead time. It must always be ready to detect an incoming request, even while its registers are read by the main processor, the DSP.

Since some wait-states for simultaneously occurring events are still inevitable, buffers have to be provided for each port for temporary storage of the data. Especially the communication between DSP and PC must be buffered well, to keep them as independent as possible.

Parallel digital I/O ports with and without line drivers should also be included in the design, to allow switching of external signals with relays or TTL signals.

PCs are in general strong radiators of noise and should be located away from the SPM. The separation of the PC and the DSP unit will also help to reduce induced noise. It is part of this project to invent a communication of DSP and PC via the standard parallel printer port of any Personal Computer system. The idea is not only to allow a convenient way of separating the devices, but also to provide compatibility to different PC systems, including Laptop computers.

Further signal improvements can be achieved by separating the ADCs and DACs from the DSP unit and by connecting them with the DSP over serial ports, as mentioned above.

Texas Instruments and Analog Devices turned out to be the best choice for 14 bit products with serial ports and programmable features. TI's device TLC320AC01 fulfills all the requirements for the medium fast purposes. It includes a programmable gains and has its own timers with adjustable offsets. The fact that it has analog in- and outputs within one single IC and that it can be linked together with ICs of the same kind allows simultaneous control of different signals, while keeping the layout dimensions small.

Analog Devices offers many high quality products for the faster purposes. A good choice is the AD7244, a fast dual DAC with serial ports.

To increase the effective dynamic range of all DAC outputs it is meaningful to keep signal offsets low and constant. This is especially important if the tip scanning plane of

constant **z** is not parallel to the surface of the SPM sample. An analog ***x***, ***y***, ***z***-coordinate rotation can help to translate the sample orientation into the **x**,**y** plane of the computer scanning area. This coordinate rotation can be achieved with a combination of several ICs of Analog Devices.

For reference purposes a summary of all setup ideas is shown in the block diagram of Appendix A.

(2.4) Concluding Comments on System Limits:

The quality of the described data acquisition system is definitely worse than the estimations made in the theoretical discussion in (2.1), since the mentioned types of noise are the theoretical error limits and do not contain non-linear behaviors of amplifiers, time drift and piezo hysteresis effects.

In general time independent non-linear disturbances are predictable and measurable and can be compensated by software algorithms.

Time dependent drift effects, however, are disturbing the measurements significantly. A compromise concerning the optimal scanning speed must be found. While on one hand slower scanning allows better filtering, it increases on the other hand the total scanning time, so that drift effects are getting more important. One possible solution is the reduction of the image size, so that less data points must be taken. Therefore each measurement can be done with the needed accuracy without exceeding a total time limit. This is particularly important if a lock-in amplifier is used to measure $\mathbf{d}I/\mathbf{d}V$ curves. The adjustable bandwidth of that amplifier influences the quality of the result significantly.

Chapter 3: Hardware Design

Before starting the design of the circuits a very strict overall guideline has to be defined. To obtain optimal system performance, it is essential to determine a clear idea of the whole setup in advance.

It is the goal to create a flexible system that can be adjusted to a large variety of experimental requirements. This makes it especially important to design an easily expandable system, so that components can be added as needed.

This purpose is achievable by creating an optimized bus architecture through which the processor unit and the system components can communicate in a well defined pattern. All main bus control circuits should be combined on one board, the DSP motherboard. This board will put all relevant signals to the pins of a single connector which will be called the Bus Connector from now on. All additional boards only have to provide an identical set of bus connectors for the incoming bus and the outgoing bus, to which the next board will connect.

The DSP Starter Kit TMS320C5x (DSK), which is used as the processor unit, is well described in TI's "TMS320C5x User's Guide" [4] and is not further discussed in this chapter. Only the facts are important, that it is programmed over the COM port of a PC and that all important signals are available on four 24 pin connectors. The on board Analog Interface occupies the built in serial port of the DSP and will be used for the output of constant reference signals.

(3.1) The DSP Motherboard:

The basic idea of any bus system is the sharing of a fixed set of signal lines to distribute data to all bus parties. So called Enable (EN) Signals are completely under the control of the DSP and determine which party is active and allowed to read or write data from or to the bus.

The TMS320C50 is a 16 Bit fixed-point DSP with 10KB on-chip memory. It offers a set of 124 instructions with 50 ns cycle time. 16 of the available 65536 I/O port addresses are accessible with fast memory mapped registers. Only those I/O ports will be used in the following design. A simple 16 channel demultiplexer (74 HC 154) transfers the binary available I/O-port-address into 16 separated Enable lines (EN00 to EN15).

Interrupt Request (IRQ) Lines allow external parties to request the right to use the bus in order to communicate with the DSP. It is up to the programmer of the DSP system to enable or disable certain IRQs and to set IRQ priorities. This technique guaranties that urgent requests are processed earlier than usual requests. Furthermore unused components can be turned off by disabling (masking) their IRQ.

Although the DSP offers some IRQ lines, the number and the timing criterias of those signals do not fulfill the requirements.

(3.1.1) The Interrupt Manager (IM):

The design of a new 8 channel IRQ-Manager (IM) is shown in Appendix B.
Each channel is connected to a flip-flop, which switches its output state if a low to high transition occurs on the input. An 8 Bit comparator (74 AS 855) compares the state of all 8 flip-flops with a previously stored value. If any incoming IRQ changes the state of the

according flip-flop, the comparator will recognize the change and notify the DSP. As soon as the DSP reads the 8 flip-flop states, the comparator stores this constellation as the reference value, so that the output of the comparator switches back to 'equal'. The next incoming IRQ will change the constellation again and switch the comparator back to 'unequal'. The DSP will be notified immediately, and so forth.

By setting the Preset Pin of any flip-flop to zero, its output state is forced to be high. As a result incoming IRQs on this line are unable to toggle the state of this particular flip-flop. The IRQ is masked.

The state of any of those Preset Pins can be determined by the DSP by writing a bitsignificant value to the Interrupt Manager (IM) address (050h). Reading the same address will give the current constellation of the IRQ-flip-flops. By comparing this value to the previously read value, the DSP can easily determine if and which IRQ occurred since the last reading.

The great advantage of this method is that even extremely short IRQ pulses can trigger the IM to notify the DSP. Furthermore the IM has no deadtime, since any IM reading by the DSP will set the IM comparator value simultaneously.

A priority list can easily be programmed to determine which events have to be taken care of before others.

The detailed schematics for this part are shown in Appendix B.

(3.1.2) The Connection to the Master PC:

Since almost any modern PC system offers a parallel Line Printer Port (LPT), the communication with the DSP will be established using this standard interface.
The pinout of this port and the I/O address of the corresponding function is shown in table 3.1.
Although the LPT is only designed for monodirectional output, it can be used for bidirectional I/O operations. The idea is to read data through the 5 handshaking input bits,

which are usually the indicators for "Paper Out", "Printer Error", "Printer Selected", "Request Acknowledge" and "Busy".

To determine when to transmit or receive which information over the port the 4 available handshaking output pins are used. The different modes are shown in table 3.2.
Missing codes can be avoided by using one of the output signals as Data Ready (DR) trigger. A pulse on this line indicates the DSP-Interface that the selected mode is valid.
DR is automatically generated, if the Init-Signal (Pin 16) is toggled. A selected mode can simply be changed by writing the value of the new mode with Bit 2 (Init) toggled to the control register of LPT. Due to the internal setup of older LPTs some modes can only be followed by certain other modes. The allowed sequences are shown in table 3.3.

The interface of the DSP is buffered to separate the DSP and the PC as much as possible. The Busy line (Pin 11 of LPT) indicates if those buffers are empty, non empty or full (see table 3.2).

Detailed information can be obtained if the Status mode is selected. The bitsignificant status value is explained in table 3.4.

If the DSP communicates with the buffers, only Bits 0 to 7 are used. After reading from the buffers, the higher bits (8 to 15) contain the status of the buffers and the recent selected LPT mode. See table 3.5 for the meaning of each bit.

The detailed schematics for this part is shown in Appendix B.

Pin	Description	I/O	Address	Bit
1	-Strobe	Output	Base+2	0
2-9	Data Bits 0-7	Outputs	Base	0-7
10	-Acknowledge	Input	Base+1	6
11	Busy	Input	Base+1	7
12	-Paper Out	Input	Base+1	5
13	-Printer Selected	Input	Base+1	4
14	-Auto Linefeed	Output	Base+2	1
15	-Printer Error	Input	Base+1	3
16	Init	Output	Base+2	2
17	-SelectIn	Output	Base+2	3
18-25	GND			

Table 3.1. Printer Port Pinout. The Standart Base Address is 0378h

Mode	Description	'Busy'-Bit
0, 4	Reset '←' buffer	unused
1, 5	Read Status	unused
2, 6	Load '←' buffer	unused
3, 7	Idle	128 if '←' buffer is full
8, 12	Reset '→' buffer	unused
9, 13	Load Bits 4-7 '→' and latch next	unused
10, 14	Reset DSP-Devices	unused
11, 15	Load Bits 0-3 '→'	0 if '→' buffer is empty

Table 3.2: Allowed Port Modes. The Mode Address is Base+2. '→' stands for DSP-to-PC transmition. '←' represents PC-to-DSP communication.

Recent Mode	Allowed Following Modes
0, 1, 2, 3	4, 5, 6, 7, 12, 13, 14, 15
8, 9, 10, 11	4, 5, 6, 7, 12, 13, 14, 15
4	0, 1, 2, 3, 8, 9, 10, 11
5	1, 3, 9, 11
6	2, 3, 10, 11
7	3, 11
12	8, 9, 10, 11
13	9, 11
14	10, 11
15	11

Table 3.3: Allowed Mode Changes.

Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Busy	**← E**	**← F**	**→ E**	**→ F**	NC	NC	NC
see tab 3.2	0 = empty	0 = full	0 = empty	0 = full			

Table 3.4: Status Byte (Base + 1) for modes 1 and 5

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
read write	**r**	**r**	**r**	**r**	**r**	**r**	**r**	**r**	**r/w**	**r/w**	**r/w**	**r/w**	**r/w**	**r/w**	**r/w**	**r/w**
use	M3	M2	M1	M0	←E	←F	→E	→F	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0

Table 3.5: The Buffer readout of the DSP. 'M' = PC-Mode-Bit. 'D' = Data-Bit (read and write possible)

(3.2) The TLC320AC01 Serial Port:

The TLC320AC01 contains an active serial interface, which provides all needed synchronization signals. It tells the DSP side of the interface when to transmit and receive data. Thus the DSP side of the interface is called passive. Since the DSP has to concentrate on many different tasks the design of the passive serial port (PSP) will contain a two level buffer of FIFO type (first in first out). Simultaneous transmission and receiving of serial data guarantees high throughput rates. Synchronizing flip-flop units are used to obtain reliable transmission of buffered data with the highest possible speed.

Different IRQ signals are generated to indicate whether one or two values have been send from the current buffer stack.

Two general PSP transmission modes are hardware selectable. For the case that the DSP stops sending data to the buffer the PSP will send the last buffered value again if the continuous mode is chosen. Otherwise, if the single-shot-mode is selected, the PSP shuts down the communication to the TLC320AC01 and will not send further IRQ signals to the DSP.

The detailed schematics for this part is shown in Appendix B.

(3.3) The AD7244 Serial Port:

The AD7244 includes two independent DACs with passive serial ports. Simultaneous conversion is only achievable by applying the same synchronization signals to both DACs. To provide those signals is the task of the active interface that has to be designed for the DSP. It is called from now on the Dual Active Serial Interface (DASI) since it provides two independent serial ports with equal synch signals. Although not used by the AD7244 both ports of the DASI can also be used to receive data.

In general two synch signals are used for serial transmission, the shift clock (CLK) and the frame synch (FS) signal. The latter will go low for exactly 16 CLK cycles during transmission of 16 bit values. Each CLK cycle will end the transmission of one bit, starting with bit 15. While data are latched out of the DASI on rising edges of CLK, data bits are latched into DASI on falling edges of CLK.

Additionally a Data OUT (DO) signal is sent to the AD7244 to trigger the output of the converted analog signals. DO is generated just before FS will go low, to latch the previously sent and meanwhile converted value out. As a result the DAC output occurs when the next value is transmitted. This is to be kept in mind when programming the devices.

The detailed schematics for this part is shown in Appendix B.

(3.4) The Parallel Digital I/O Port:

This little board just simply provides some line drivers with one level buffers for two of the I/O addresses of the DSP. While one of them is designed for TTL data in and output, the other address provides power outputs for external relay switches. For testing purposes a third output driver is provided to connect a 14 segment LCD display with two digits.

The detailed schematics for this part is shown in Appendix B.

(3.5) The Analog Boards:

The analog boards are designed very simply. Their dimensions are kept small so that they fit into the UHV chamber of the SPM.

Since all required handshaking is done on the digital boards, the analog circuits are basically just connecting the ADCs and DACs with the incoming signal plugs. Outgoing

analog signals are preamplified and impedance matched on the board itself. Large grounded copper areas are placed on critical positions to shield analog signal wires from digital signals. The PCB layouts of those boards are shown in Appendix C.

The board which is designed for the coordinate transformation consists of a combination of amplifiers, multipliers and sin(x) generators. It realizes the following mathematical equations:

$$x' = x_{gain} \cdot x \cdot \cos\alpha + y_{gain} \cdot y \cdot \sin\alpha + x_{ofs}$$

$$y' = -x_{gain} \cdot x \cdot \sin\alpha + y_{gain} \cdot y \cdot \cos\alpha + y_{ofs}$$

$$z' = z_{gain} \cdot z + x_{slope} \cdot x' + y_{slope} \cdot y' + z_{ofs}$$

where

$x, y, z, x_{ofs}, y_{ofs}, z_{ofs}, x_{gain}, y_{gain}, z_{gain}, x_{slope}, y_{slope}$ and α

are the input signals.

Chapter 4: The Final Assembly

Once the schematics are translated into the final layout, PCB boards can be manufactured. Although it is a major task to create and to test such layouts and to assemble all electrical devices, it is not part of this work to describe the single steps.
The individually assembled and tested digital boards are finally added up to a complete stack of parties on the bus system. The stack order is shown in figure 4.1.



Figure 4.1 Stack Order of Bus Parties

For shielding purposes the complete stack of digital boards, including the power supply is mounted in a closed aluminum box. A small internal fan circulates the enclosed air, to help distribute locally generated heat towards the walls of the box, which will radiate it to the environment. This is enough to keep the system from overheating. However, the power supply shows temperature related problems after about half an hour. A secondary, external fan is recommended to be utilized during idle times of the system, i.e. programming of the DSP or data processing of the PC. This secondary fan should only be turned off for real measurements, when high accuracy is needed.

For future references the outgoing connectors of the digital box are shown in figure 4.2. The names refer to the previously discussed boards, where DSP represents the DSP motherboard. The front panel of this box provides the LPT and COM connectors and a 14 digit LCD display that can be accessed by the programmer for software testing and debugging purposes as well as for mode display features.

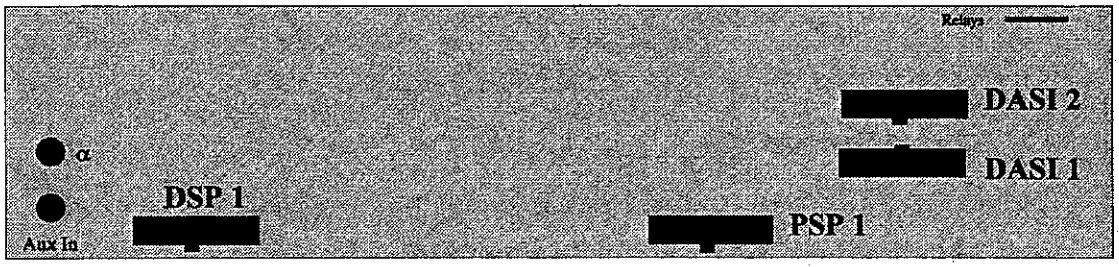


Figure 4.2 The Back Panel of the Digital Box.

Unlike the digital boards, the analog circuits need to be individually shielded with little closed metal boxes. Those boxes are carefully linked with shielded twisted wire pairs. All internal analog signals are balanced signals, which means that one wire of the twisted pair is exactly the negative signal of the other. Thus crosstalk and noise induction are kept low.

As long as those boxes are not placed directly within the vacuum chamber they are put all together into an aluminum box for further shielding.

BNC connectors are used for the final output of all analog ground-referenced signals. The front panel of the box is displayed in figure 4.3. The system can be used to drive both, tripot and tube scanner devices. The latter is selected by switching the shown switch to "**w = z**".

All scanner outputs and walker signals (w±x, w±y, ±z, WalkerL, WalkerH) are capable of driving up to 100 mA load with ±14 V voltage swing. Scan amplitudes and slope rates can be adjusted with 1% accuracy anywhere between -1 and +1 of the computer generated signals. Unfortunately the used potentiometers turned out to be the wrong choice, so that especially at high signal frequencies inductive effectes are recognizable.

To complete the documentation of the final assembly, figure 4.4 is added. It shows the back panel of the analog box. The connectors are named according to the previously mentioned definitions. The final addresses and IRQ lines are noted in this figure as well. They are determined by the order of how the digital components are strung together. Thus if this order is changed, those addresses will change accordingly.

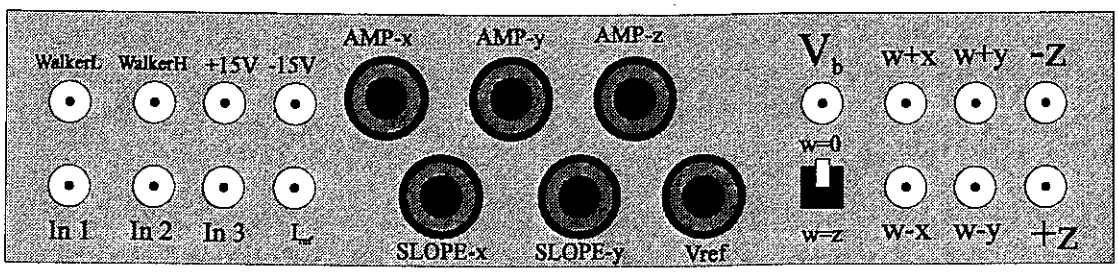


Figure 4.3 Front Panel of the Analog Box.

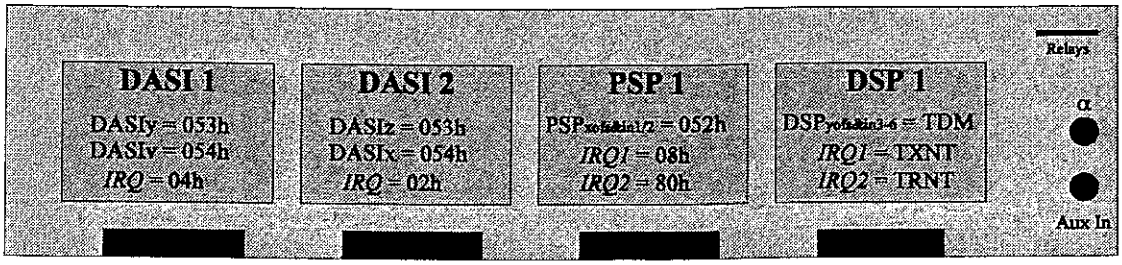


Figure 4.4 Back Panel of the Analog Box.

Chapter 5: The Software

(5.1) TI's Assembler:

Texas Instruments delivers the TMS320C50 Digital Signal Processor Starter Kit (DSK) with very simple but extremely useful assembler and debugger programs. Those can be used to convert assembly language source code into a format that is understood by the driver software that comes also with the DSK. It is the task of those drivers to transfer the DSP programs to the DSK and to start them.

The debugger is a neat tool to analyze simple programs. However, it is not a good help if real time processes are handled by the DSP. In this case the Debugger is just too slow to allow the DSP to run properly. Unfortunately most of all SPM controlling purposes have to be done in real-time. The only possible way of analyzing those programs is the usage of the LCD display of my hardware design. Its DSP address is 0059h and can be easily accessed by the programmer.

In general assembly language is not that hard to learn as it might appear to a beginner. Its small set of instructions makes it fairly easy to get a good overview of the processor's abilities. Some insights into the design of the electronics are of course necessary to be able to use this language properly.

Due to its limited set of instructions, programs are very abstract and hard to read if they are not well documented. To improve the readability it is extremely helpful to divide the program into separated function groups. Furthermore this technique helps to trace errors. However, the larger the number of program parts, the more time consuming is the editing and changing of the program. It would be very helpful to have a software tool that allows fast and easy editing and switching between program parts.

(5.2) The DSK Programmer Tool:

The motivation is to create a new PC Software Tool that covers within one single program all main features like selecting, editing, compiling and starting of DSK programs. This will allow effective and fast writing of DSP software units.

Once this task has been established, the PC software can be extended to include further options, like communication to the DSP over the printer port and final data processing.

The communication with the DSP depends on the standards which are defined by the program that has been started in the DSP. Thus it is meaningful to define a short set of general instructions that allow the PC complete control over the DSP. As soon as a standard of communication is defined, the PC only has to send a short list of numbers to tell the DSP which of the programmed options are to be executed with which parameters.

If the DSP program is written in a well organized structure, future options can be added easily.

It should be the goal of each module to be able to run independently of the PC. This helps to keep the PC available for other purposes. The PC has now the option, to let the DSP run completely self-controlled which means that the DSP executes along the programmed line of modules. However, the PC can always interrupt and choose between three basic options:

1) execute next module, 2) execute until a certain event occurs and 3) execute a single module that is not the next in line.

For general SPM image taking control, it is important to establish auto approach and scanning procedures. If the feedback is not done by analog circuits a corresponding algorithm has to be provided as well.

The auto approach is done completely digitally. It consist of three modules:

1) Withdraw the tip with a smoothly generated ramp.
2) Step the walker one step forward
3) Expand the z piezo drive while constantly checking the incoming signal. In the case of a STM the tunneling current is monitored.

As long as no tunneling current is detected, the DSP will follow those three steps in a circular motion. Once tunneling occurred the scanning mode is activated. It consists of basically two modules which let the tip scan one step at a time in either the x or y direction.

If digital feedback is provided, the x and y positioning has to be done very smoothly with constant monitoring of the incoming signal, i.e. the tunneling current.

While scanning along the surface of the sample, the DSP transmits pre-processed image data to the PC. Meanwhile the PC is displaying and storing all information.

Thanks to *Günter Guttroff* I did not need to spent much time on writing display procedures for the PC. Günter kindly let me include parts of the object code of one of his data processing programs.

All available DSP instructions are mentioned in table 5.1. The precise usage can be read out of the source code of the program, which is partly printed in the appendix. Only selected procedures could be added to the appendix since the complete source code of the PC program covers almost 100 pages in small letters. Nonetheless a complete printout of the DSP program is included in the appendix, since it is very essential for future references.

Another program module has not yet been mentioned. The following chapter will therefore concentrate on the topic of dynamic range expansion.

Instruction Code	Description	Instruction Code	Description
00h	Halt any process	stuv14h	Set Walker Amplitude
88h	Continuos mode	stuv24h	Set Dots per Cycle
77h	Next Step	stuv34h	Set I_{ref}-hi for Approach
y9h	(See Table 5.2)	stuv44h	Set I_{ref}-range (Approach)
16h	Walk left	stuv54h	Set I_{ref}-hi for Scanning
26h	Walk forward	stuv64h	Set I_{ref}-range (Scanning)
36h	Walk right	stuv74h	Set x Step Size
46h	Walk backward	stuv84h	Set y Step Size
56h	Contract z	stuv94h	Set Bias Voltage
66h	Expand z with Read	stuvA4h	Set x Position
67h	Scan x until end of line	stuvB4h	Set y Position
68h	Scan Bias Voltage	stuvC4h	Set Bias Voltage Step Size
stuv5h	Set Step counter	stuvF4h	Set z Position

Table 5.1 Instructions.

Instruction Code	Execute until	Instruction Code	Execute until
19h	Contract z	59h	Scan x
29h	Walk	69h	Scan until End of Image
39h	Expand z with Read	79h	Scan until z Out of Range
49h	(used for PSP Setup)	D9h	Scan y

Table 5.2: 'Execute until' Instruction. The DSP will stop before the chosen Event will be executed.

(5.3) Dynamic Range Expansion:

First done by Feenstra *et al.* [2], the method of expanding the dynamic range of the acquisition system by simultaneous z and V scanning still bears mathematical problems.

The idea is simple in principle. The detected tunneling current I_t of an STM experiment follows roughly a power law dependence on the applied bias voltage V_b. Thus if V_b is varied over a large range, I_t will change dramatically. Resolution problems are unavoidable unless efforts are taken to boost up small current signals. For this purpose Feenstra could establish a very precise tip movement that pushed the tip closer to the sample while V_b was decreasing. This method takes the exponential z dependence of I_t into account which will amplify the detected signal if z is decreased. The applied V and z signals are shown in figure 5.1

Problems occur as soon as physically relevant results need to be drawn from the obtained set of data. To be able to calculate band energy profiles for the studied sample all tunneling information must refer to a constant separation z_0 of tip and sample. Since all data points were taken with simultaneously changed z and V values it is impossible to compare any two data points with each other. A very precise $I(V,z)$ relation is thus necessary to refer all data to the reference position z_0. Unfortunately this relation is not yet known perfectly. Only estimates were done which are assumed to be equal for all x,y positions of the scanning area. The quality of the derived physical conclusions thus depends significantly on the accuracy of this estimate.

In the following a different method is introduced that describes a new technique of expanding the dynamic range of the acquisition system based on the same physical background.

Assuming that the tunneling current is a very low noise signal, methods of linear algebra can be used to compute a polynomial relation between two sets of data, which were taken for identical V_b values, but at two different z levels. In contrast to the previously described

method, z is held constant for each set of data. The applied V and z signals are plotted in figure 5.2.

The idea is to let the feedback move the tip to a start position z_0, before it is stopped. Then V_b stepping is started, until I_t drops below a determined threshold. The feedback is turned on again to bring the current back into range. The new z position is z_1 and the feedback is stopped again. V_b will be changed back by N steps and scanning will be continued. Thus the first upcoming N data points will have identical V_b values as the last N data points taken at z_0. A polynomial P_1 of order $(N-1)$ can be computed, that relates those N $I_t(z_1,V)$ values back to the $I_t(z_0,V)$ values taken before. If V_b scanning is continued, further data points $I_t(z_1,V)$ can be taken until I_t drops again under the determined threshold. While taking those data the DSP can calculate in real-time what the according $I_t(z_0,V)$ data would have been, just by using the polynomial relation P_1. The PC will thus receive immediately the computed information, all referenced to the same position z_0.

The feedback can be turned on again to bring I_t in range. z_2 is the new position where the feedback is stopped. V_b is steeped back, scanning will be started and a new polynomial P_2 can be determined to relate z_2 values back to z_1. Since P_1 is still known, z_2 values can be related even back to z_0 by combining P_2 and P_1. This can be done in one step, if P_2 is calculated by using the already z_0 related z_1 values in this computation.

This method can be repeated several times, before V_b scanning is finished.

It is not required to send any z-position information to the PC.

No problems will occur as long as the tip positioning is stable enough and the tunneling current steady. However, if too many z steps are taken then involved errors of the polynomial fitting will increase towards the end of the scanning, since all values at z_j are related to z_0 by using all previous polynoms P_1 to P_j. Thus a very careful choice of N and the I_t threshold is essential.

If the feedback is not used to reposition the tip, but if equal sized z steps are taken, then a further improvement is possible. Since the tip has to move first towards the sample and then, after V_b crosses zero, away from the sample, there will be identical z -positions for two different V_b ranges. If hysteresis and drift effects of the piezo drivers can be neglected then the related polynomials should be identical for those two V_b ranges with same z level. This can be taken into account to eliminate parts of the involved mathematical error.

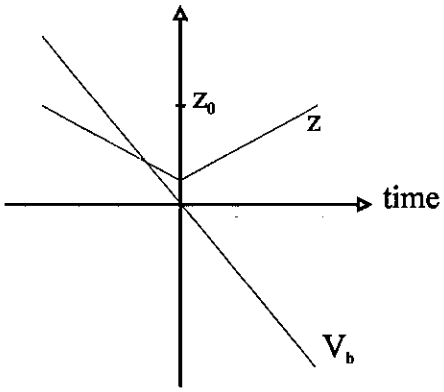


Figure 5.1 Feenstra's method.

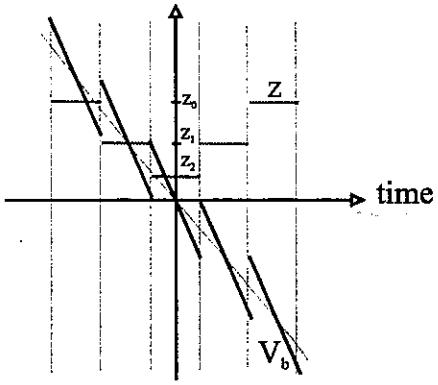


Figure 5.2 Tasler's method.

The calculation of the polynomial is very simple using matrices. Assuming $N=6$, P_1 will be of the order 5. The second set of data is called x_1 to x_6 and needs to be referred to the first set, called y_1 to y_6. The equation for x_1 is:

$$P_1(x_1) = a + bx_1 + cx_1^2 + dx_1^3 + ex_1^4 + fx_1^5 = y_1$$

By writing also all other equations down, we will obtain the system matrix S:

$$S := \begin{bmatrix} 1 & x1 & x1^2 & x1^3 & x1^4 & x1^5 \\ 1 & x2 & x2^2 & x2^3 & x2^4 & x2^5 \\ 1 & x3 & x3^2 & x3^3 & x3^4 & x3^5 \\ 1 & x4 & x4^2 & x4^3 & x4^4 & x4^5 \\ 1 & x5 & x5^2 & x5^3 & x5^4 & x5^5 \\ 1 & x6 & x6^2 & x6^3 & x6^4 & x6^5 \end{bmatrix}$$

A mathematical rule says, that the coefficient ***a*** is expressed by:

$$a = \frac{\det[A]}{\det[S]}$$

where ***A*** is formed, by replacing the first column of ***S*** with y_1 to y_6

$$A := \begin{bmatrix} y_1 & x_1 & x_1^2 & x_1^3 & x_1^4 & x_1^5 \\ y_2 & x_2 & x_2^2 & x_2^3 & x_2^4 & x_2^5 \\ y_3 & x_3 & x_3^2 & x_3^3 & x_3^4 & x_3^5 \\ y_4 & x_4 & x_4^2 & x_4^3 & x_4^4 & x_4^5 \\ y_5 & x_5 & x_5^2 & x_5^3 & x_5^4 & x_5^5 \\ y_6 & x_6 & x_6^2 & x_6^3 & x_6^4 & x_6^5 \end{bmatrix}$$

The coefficient ***b*** is calculated analogously to ***a***. The matrix ***B*** is created by replacing the second column of ***S*** with y_1 to y_6.

Also the coefficients *c* through *e* can be computed using the same technique.

While the system determinant can be simplified:

$$\det[S] = \prod_{i=2}^{N} \prod_{j=1}^{i-1} (x_j - x_i)$$

it is not possible to simplify *A*, *B*, *C*, *D* and *E* in general. However, the calculation comes down to a relatively small number of multiplications and summations. The DSP is able to compute a product and a sum in 50 ns, so that the calculation is done in several milliseconds.

Chapter 6: Results and Conclusions

This work was able to discuss and design a new setup of a universal SPM control and data acquisition unit. Furthermore all schematics were transformed into PCB boards, tested and finally assembled. A unique and flexible way of communication between the new system and any PC is established and a software tool written that allows easy programming and executing of DSP programs with an IBM compatible PC.

A digital feedback algorithm is also included and successfully tested. Not yet finished is the program described in Chapter 5. However, parts have already been tested and appear to work as predicted.

Unfortunately no high quality STM was available during the last weeks of this project, so that not even preliminary data could be taken. However some studies with an old STM, which is not yet optimized, could be achieved. Although the low frequency components of the initial noise of that system exceeded 15% of the measured signal amplitude, first rough images of a graphite sample could be taken.

The result is shown in figure 6.1. Although the patterns look like noise effects, it appears to be a real structure. The image contains 256*256 data points, each of which took 0.45 seconds to be recorded. This seems to be unusually long but is explained by the averaging that has to be done to filter the immense noise. The change in the pattern of the image is apparently related to vibration noise that displaced the tip permanently.

The future goal is to set up the above described control system with a highly optimized STM. To obtain the resolution of electronic states of an GaAs/AlGaAs/GaAs quantum well a low temperature STM in UHV is necessary. The program is already capable of taking topographic images. All parameters, like bias voltage and reference currents can be set by software. Thus an adjustment of the program should be practicable, especially if an STM experienced person can provide the information about optimal settings.

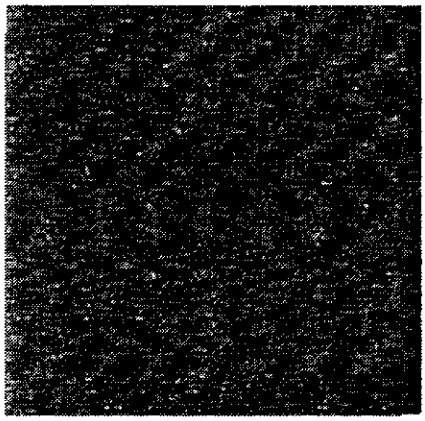
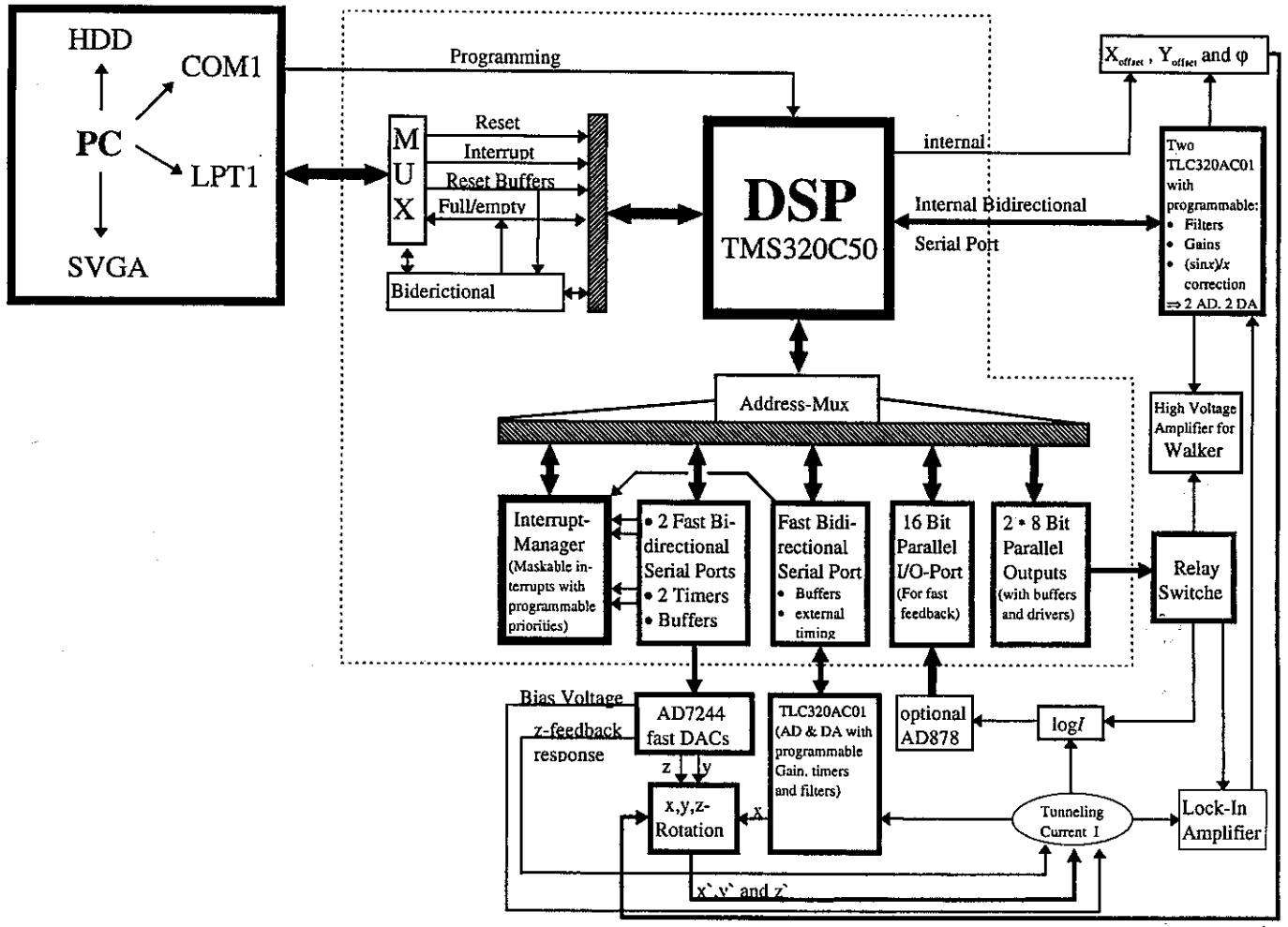


Figure 6.1 First Graphite Image.

Appendix

(Appendix A): Block Diagram of SPM Control Unit



(Appendix B): Schematics:

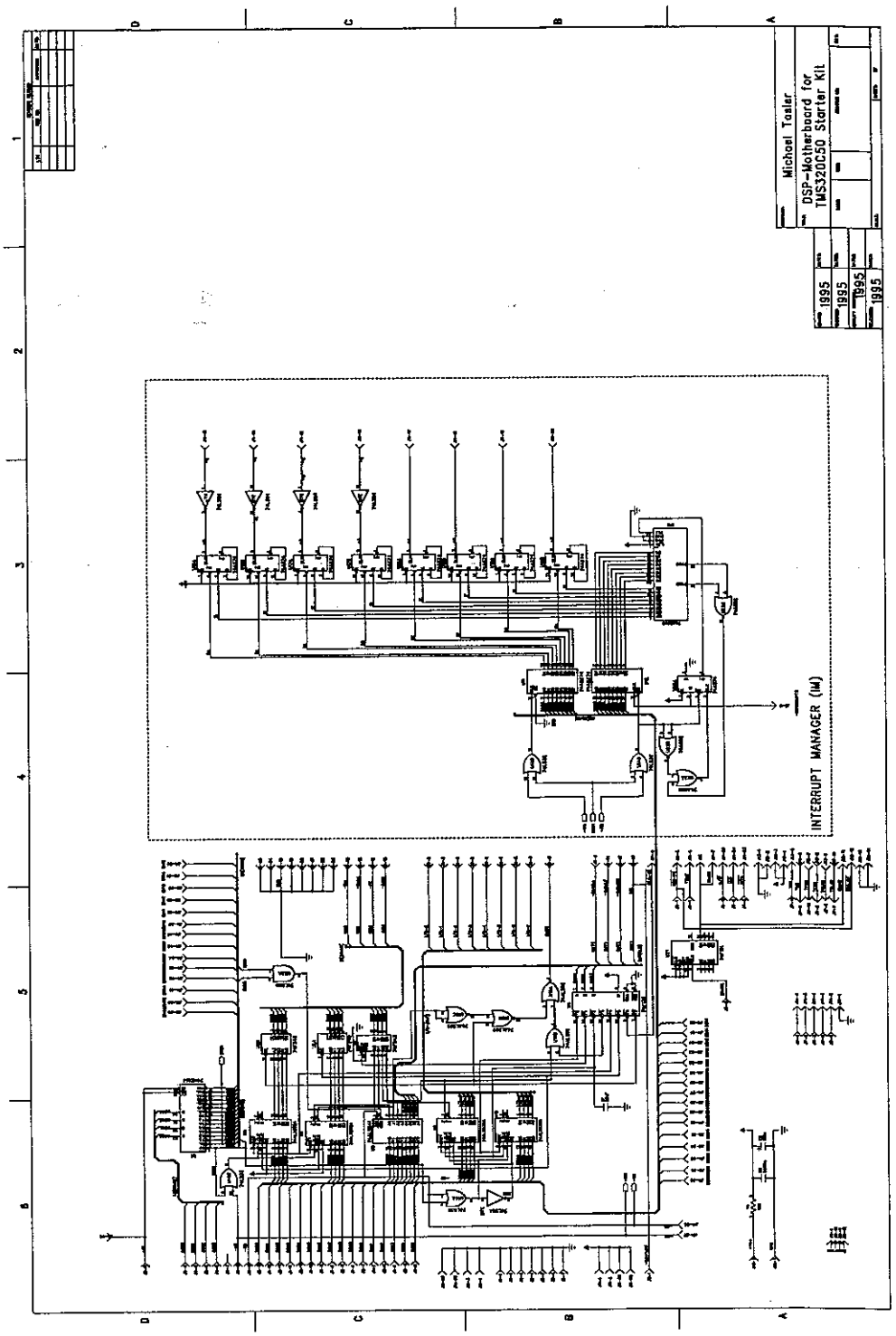


Figure B.1: Schematic of the DSP-Motherboard

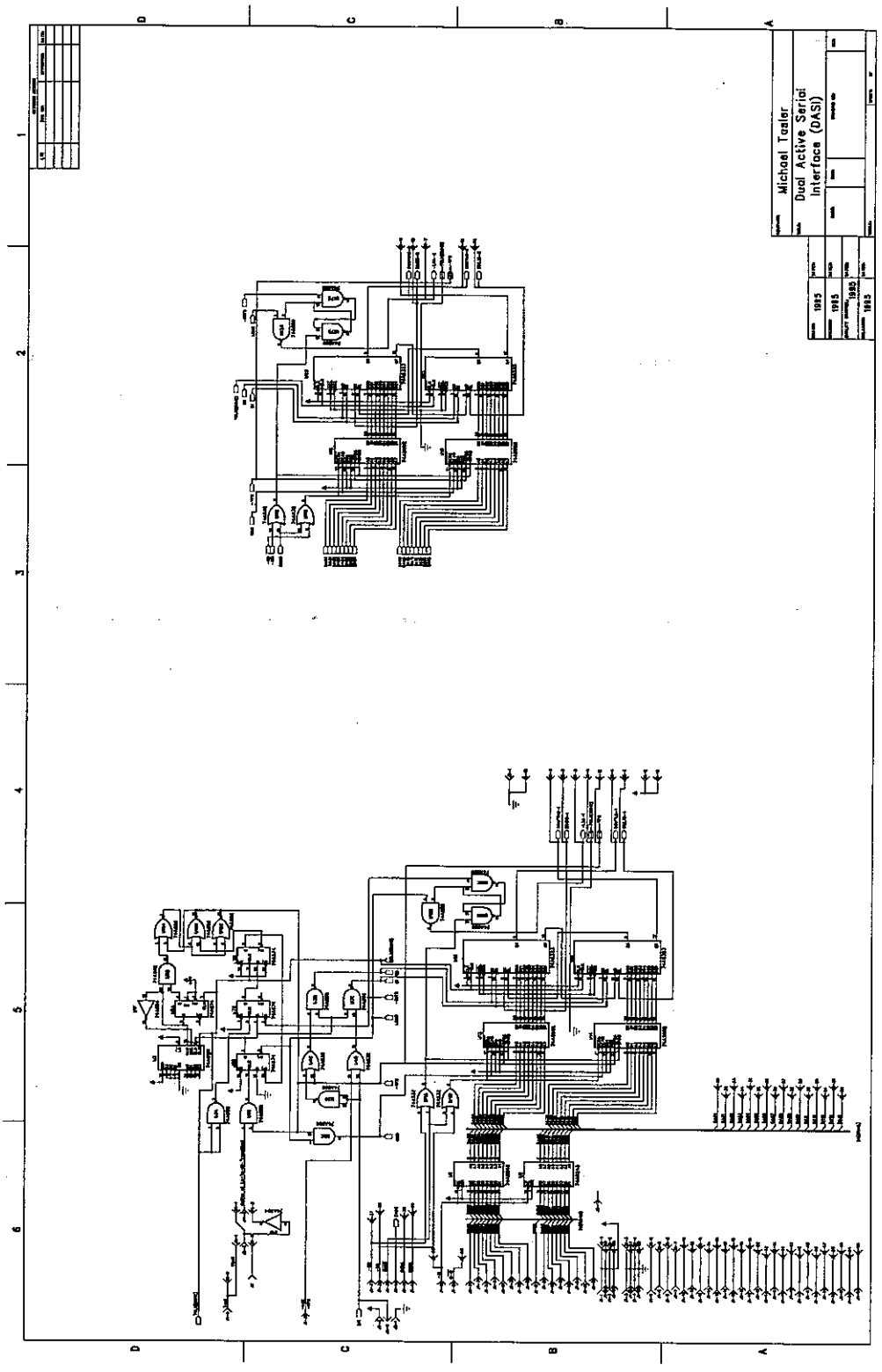


Figure B.2: Schematic of DASI

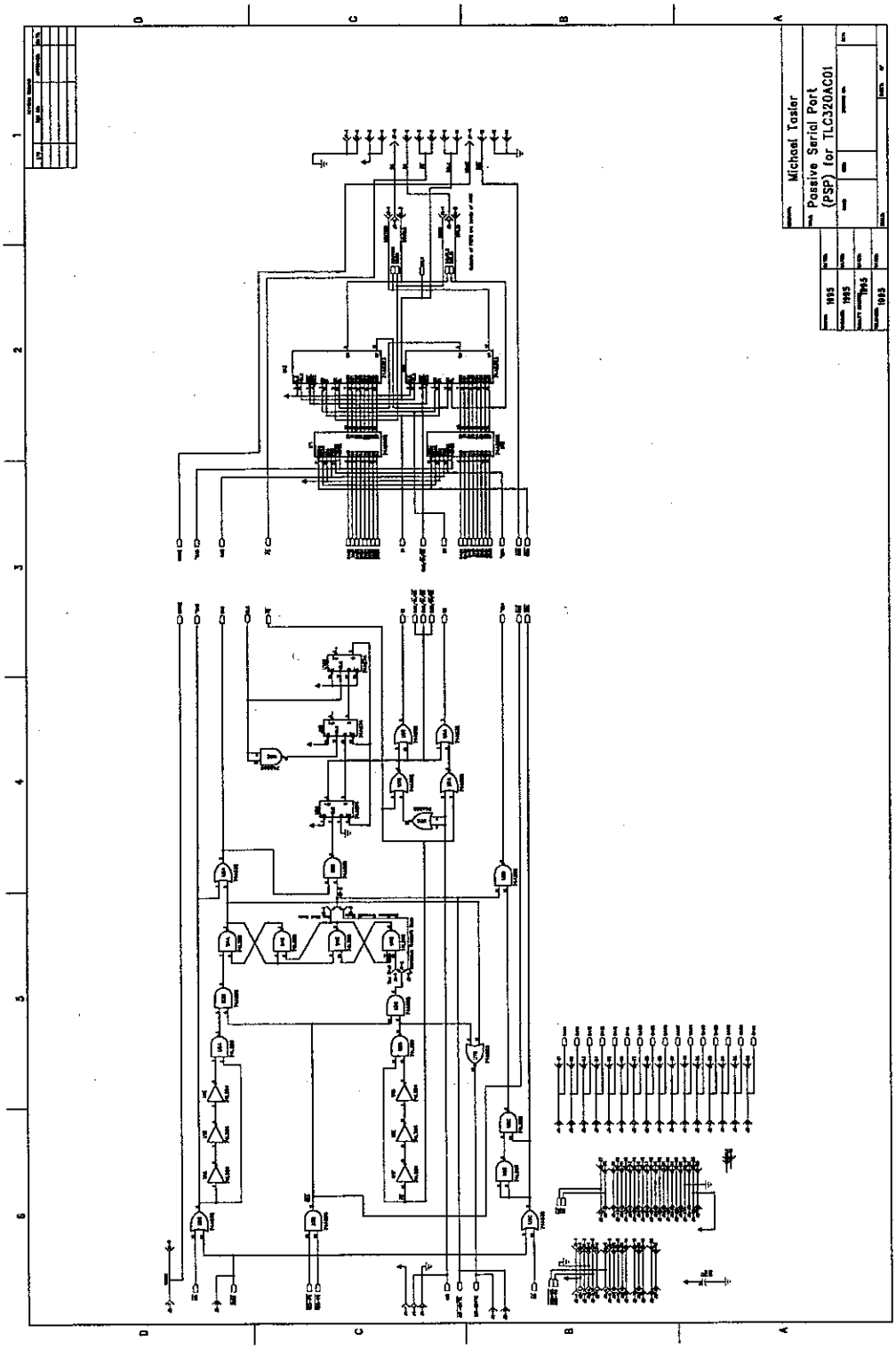


Figure B.3: Schematic of PSP

(Appendix C): PCB Layouts:

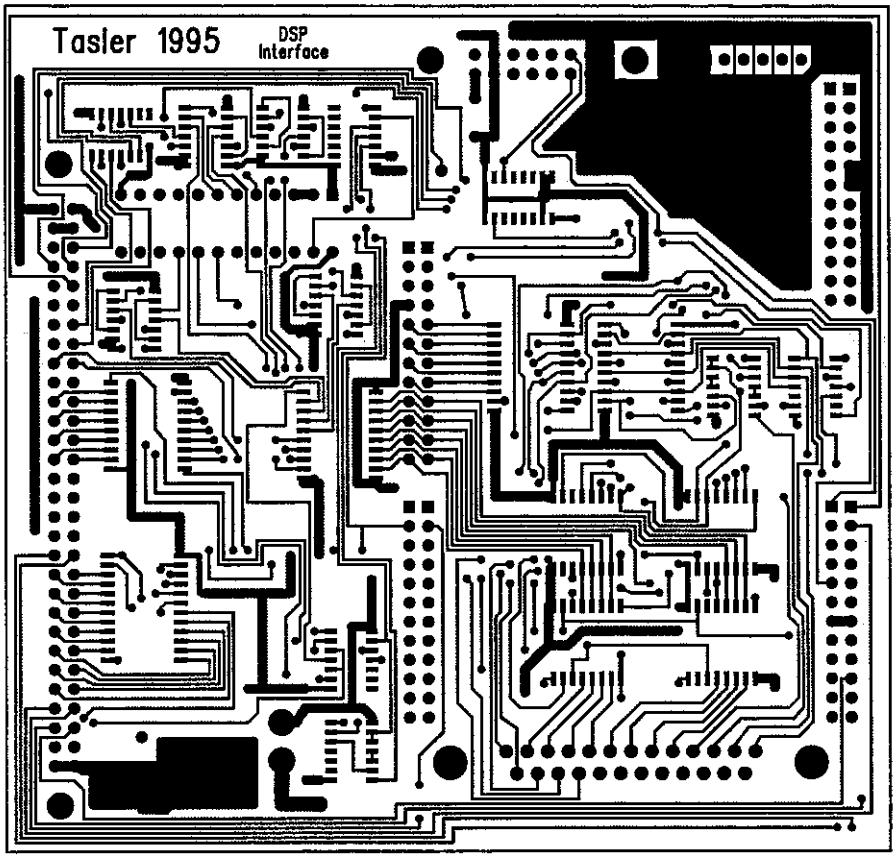


Figure C.1.1: PCB Layout of the DSP Motherboard (Component side)

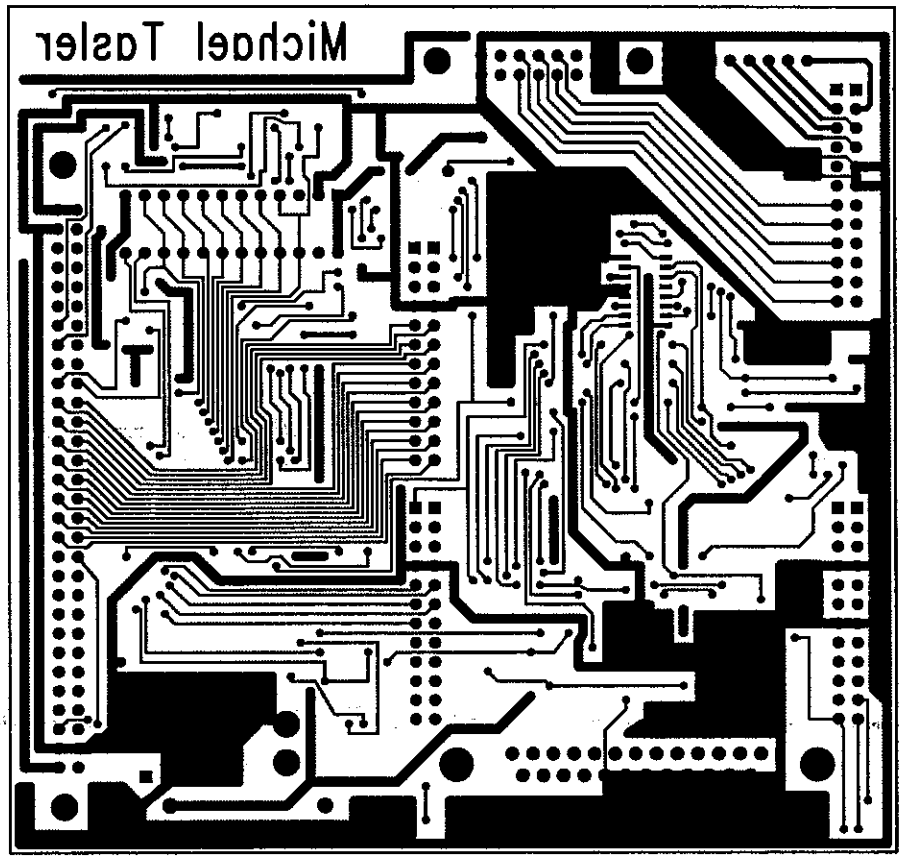


Figure C.1.2: PCB Layout of the DSP Motherboard (Solder side)

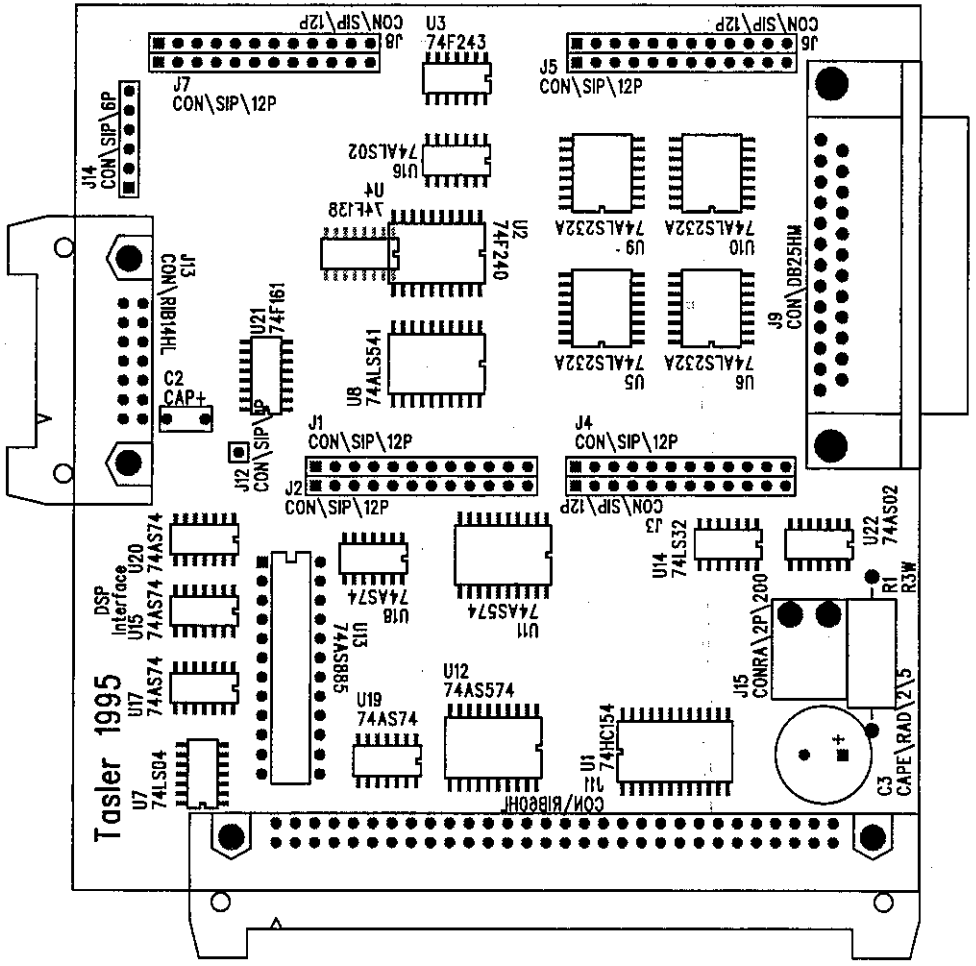


Figure C.1.3: PCB Layout of the DSP Motherboard (Components)

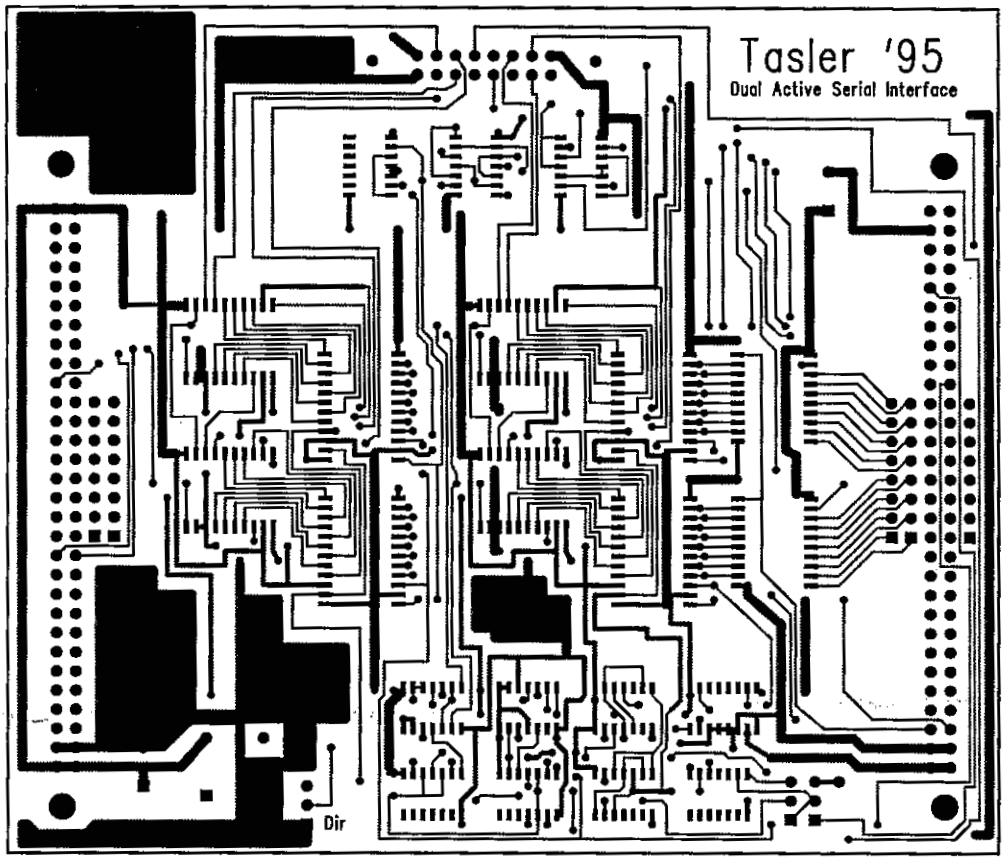


Figure C.2.1: PCB Layout of DASI (Component side)

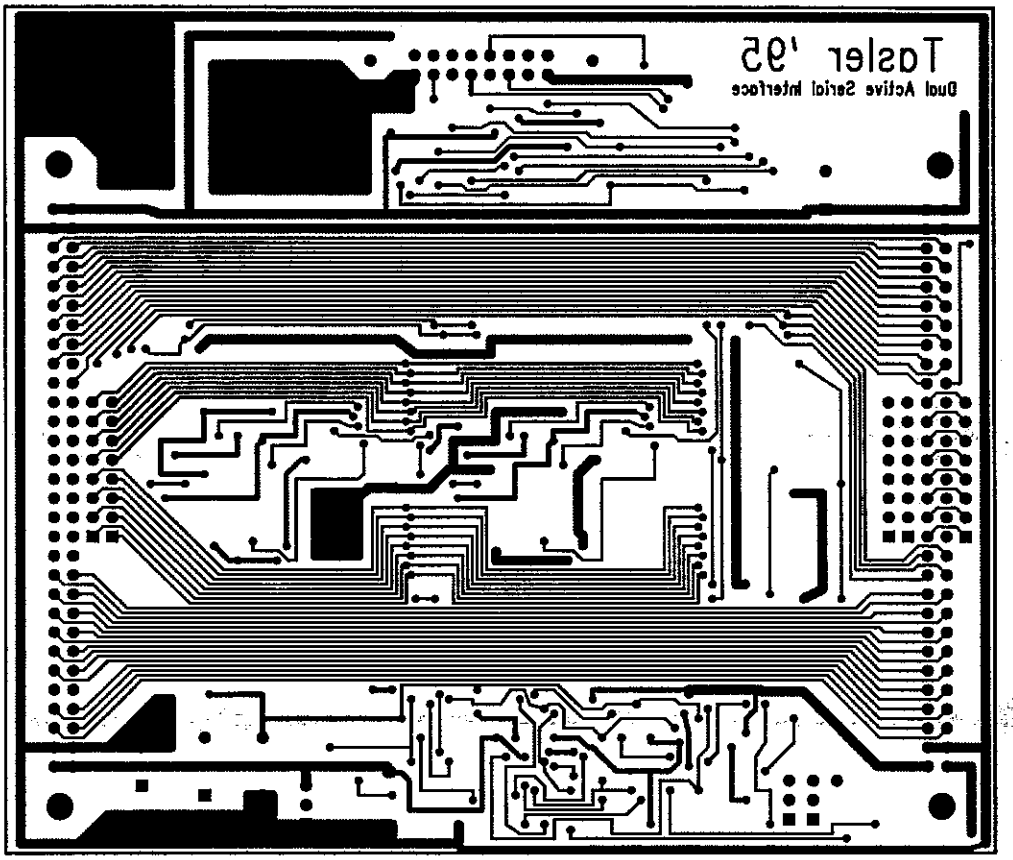


Figure C.2.2: PCB Layout of DASI (Solder side)

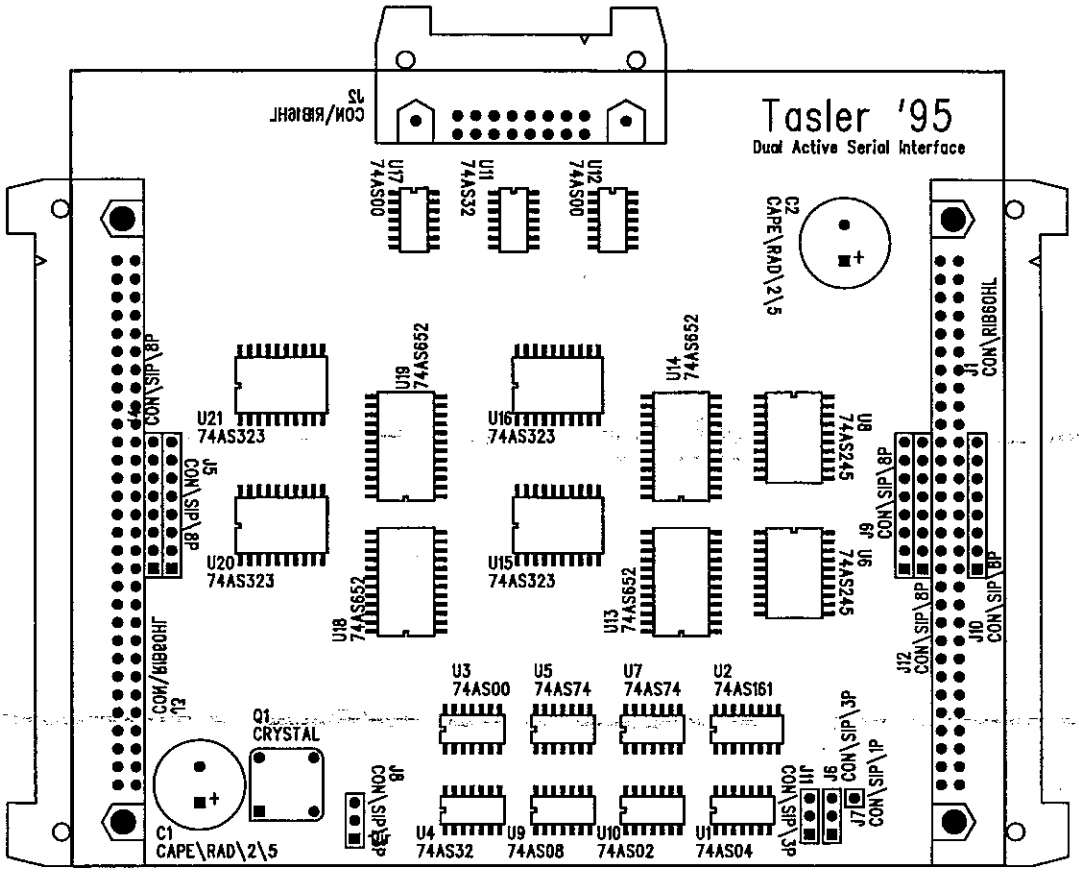


Figure C.2.3: PCB Layout of DASI (Components)

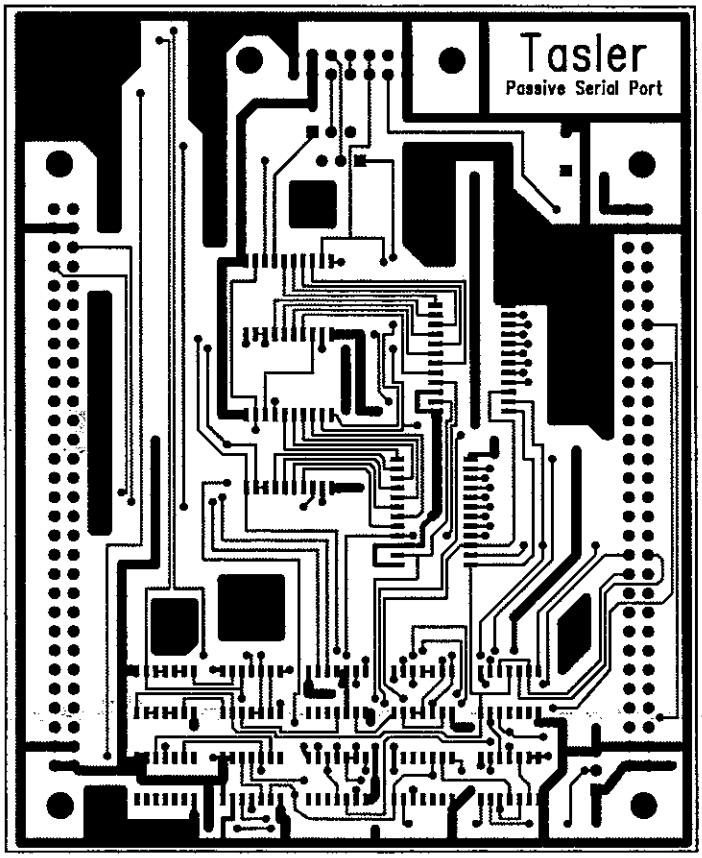


Figure C.3.1: PCB Layout of PSP (Component side)

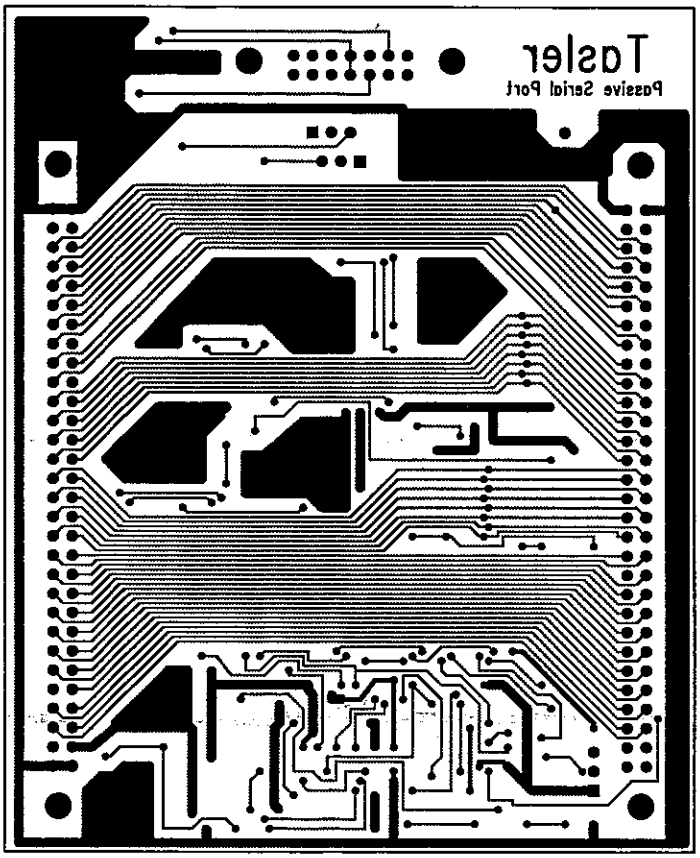


Figure C.3.2: PCB Layout of PSP (Solder side)

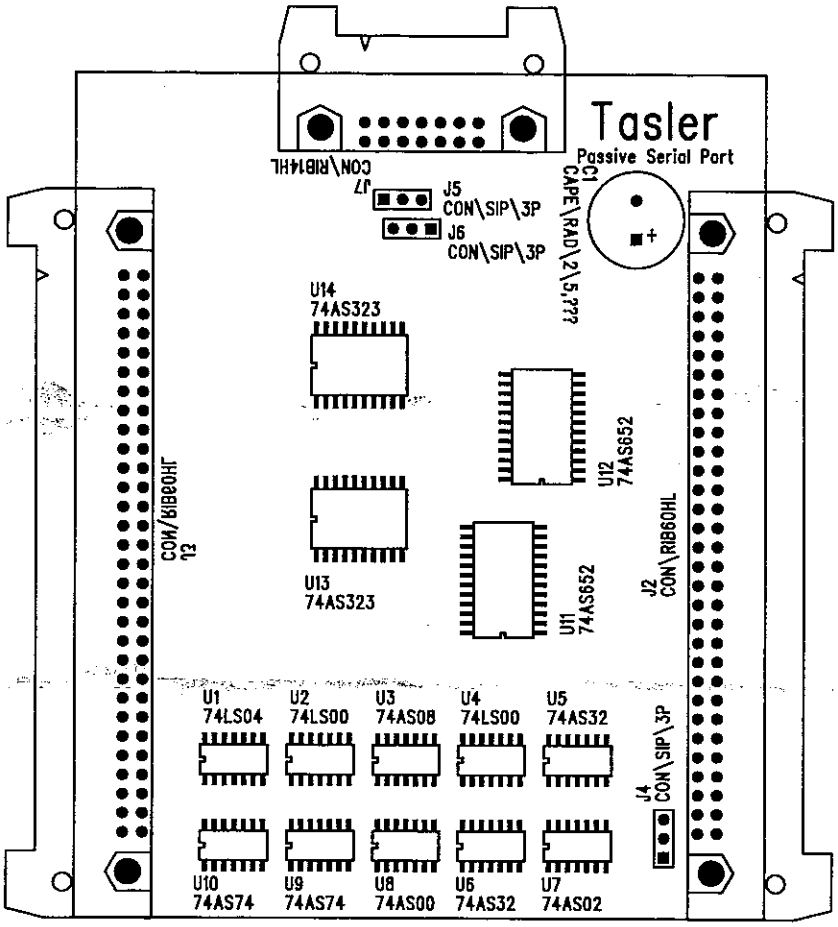


Figure C.3.3: PCB Layout of PSP (Components)

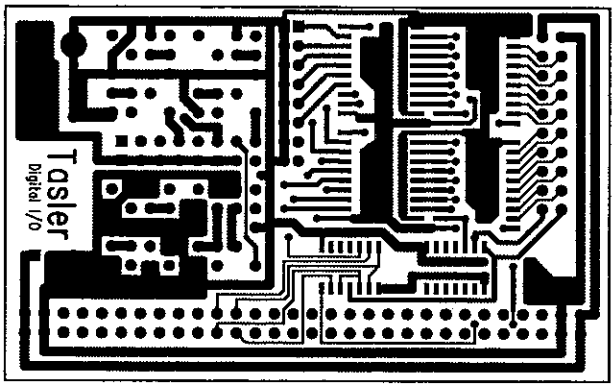


Figure C.4.1: PCB Layout of PIO (Component side)

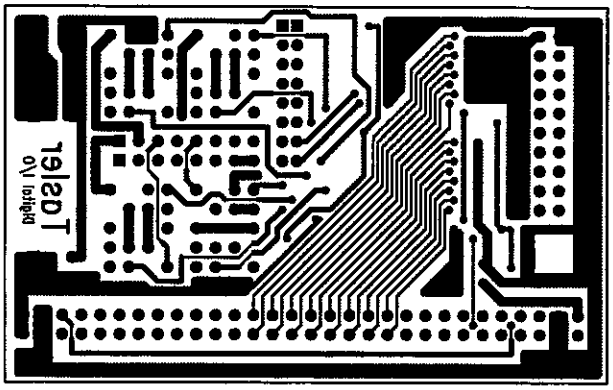


Figure C.4.2: PCB Layout of PIO (Solder side)

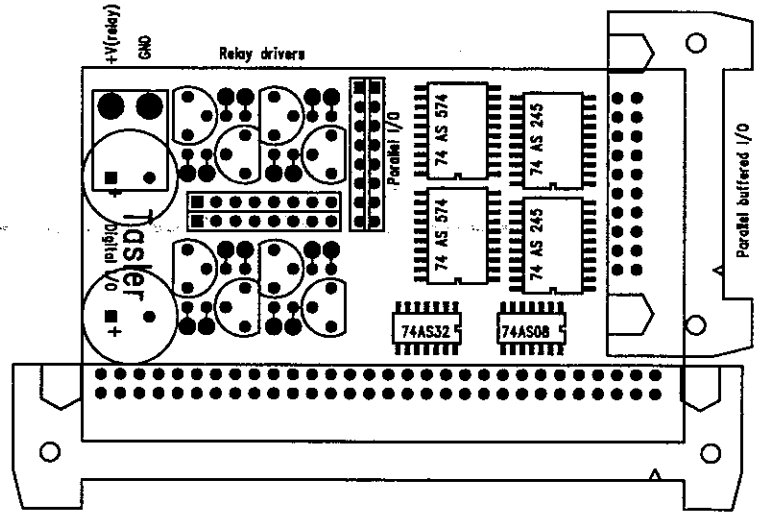


Figure C.4.3: PCB Layout of PIO (Components)

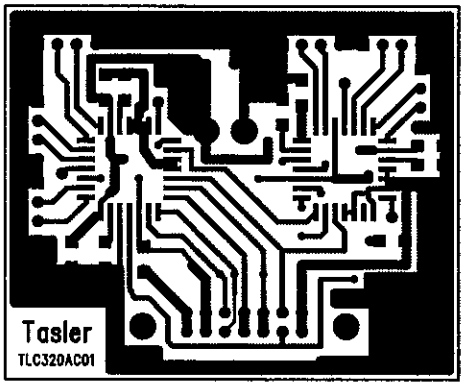


Figure C.5.1: PCB Layout of TLC320AC01 (Component side)

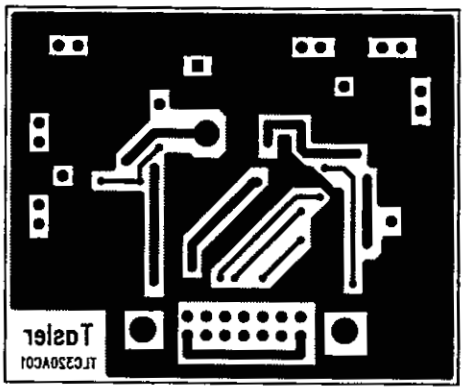


Figure C.5.2: PCB Layout of TLC320AC01 (Solder side)

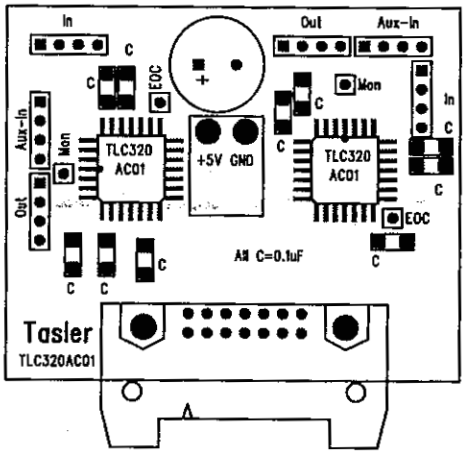


Figure C.5.3: PCB Layout of TLC320AC01 (Components)

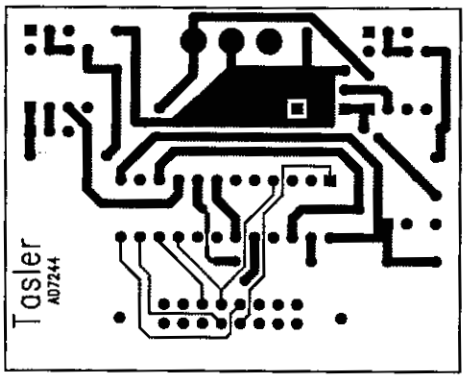


Figure C.6.1: PCB Layout of AD7244 (Component side)

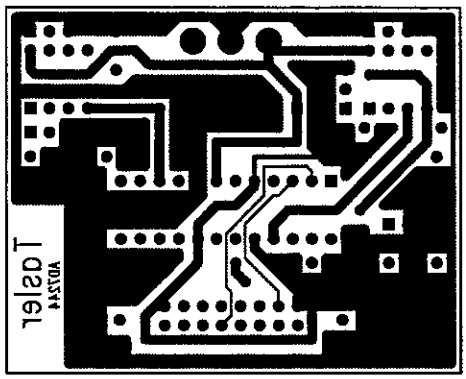


Figure C.6.2: PCB Layout of AD7244 (Solder side)

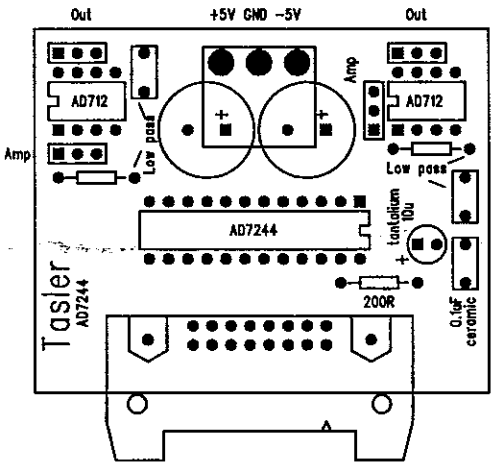


Figure C.6.3: PCB Layout of AD7244 (Components)

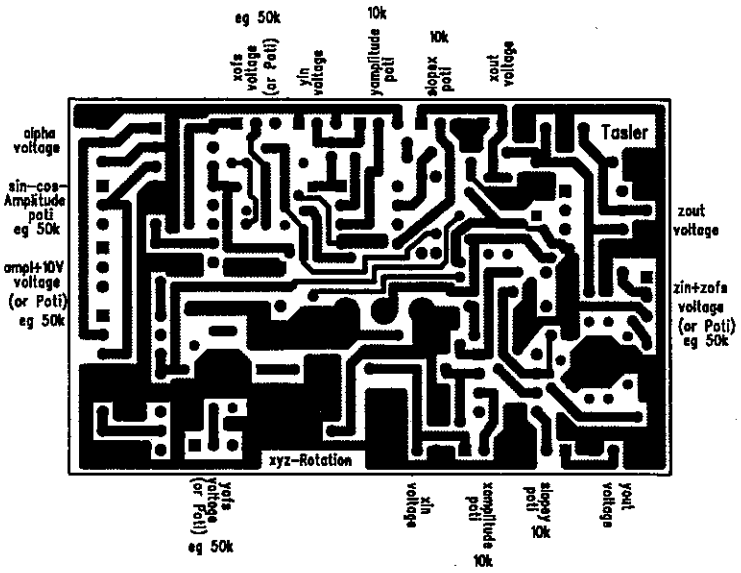


Figure C.7.1: PCB Layout of Coordinate Rotation (Component side)

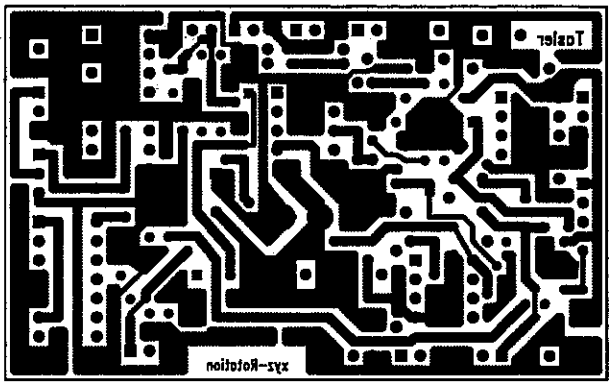


Figure C.7.2: PCB Layout of Coordinate Rotation (Solder side)

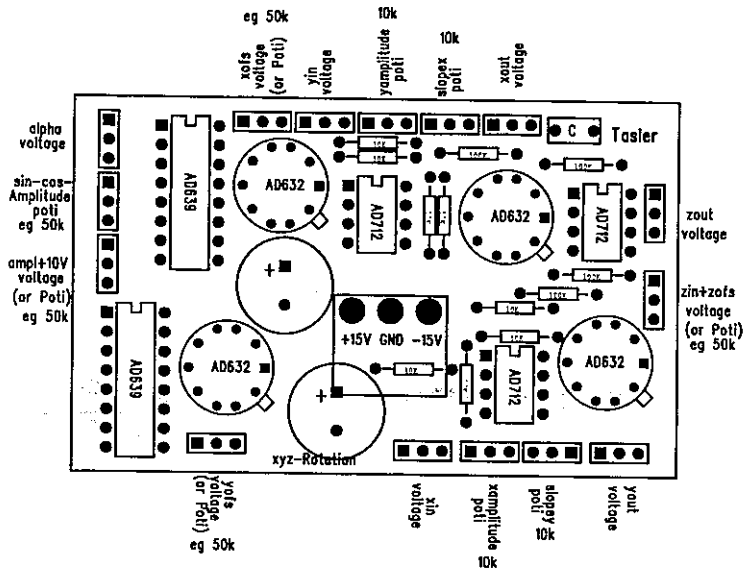


Figure C.7.3: PCB Layout of Coordinate Rotation (Components)

(Appendix D): The PC Software:

This appendix contains the source code of selected C++ procedures of the software tool, which is discussed in Chapter 5.

The PC to DSP Comunication Procedures:

```
int outinstruction(int inst1, int inst2, int inst3)
{ // 0 is transfered to 1,1 while 1 is transfere to 1,2
 unsigned char readback[21];
 unsigned char check[20];
 unsigned char check2[20];
 int giveback=0;
 int cnt=0;
 int cnt2=0;
 int a1,a2,a3,a4;
 int i;
 char *ptr;
 inst1=inst1&15;
 do {
  do {
   strcpy(readback,outvalue(inst1));
   check[0]=129;
   check[2]=0;
   check[3]=0;
   if(inst1>1) check[1]=inst1;else {check[1]=1;check[2]=1+inst1;}
   ptr=strstr(readback,check);
   if(!ptr)
   {//not found
    cnt=0;
    check[0]=255;
    check[1]=255;
    check[2]=1;
    check[3]=1;
    check[4]=0;
    while((cnt<6)&&(!(ptr=strstr(outvalue(0),check)))) cnt++;
    if(cnt==6) giveback=-1;
   } else giveback=1;
  cnt2++;
 } while((giveback==0)&&(cnt2<3));
 i=0;
 if(giveback>0)
 { // here inst1 is send successfully.
  ////////////
  if(inst1!=5)
  {
   giveback=0;
   strcpy(readback,outvalue(inst2));
   if(inst1!=4)
   {
    check[0]=255;
    check[1]=255;
    check[3]=0;
    check[4]=0;

    if(inst1>1) check[2]=inst1;else {i=1;check[2]=1;check[3]=1+inst1;}

   } else
```

```
    {
     check[0]=126;
     if(inst2>1) check[1]=inst2;else {check[1]=1;check[2]=1+inst2;}
     check[2]=0;
    }
    ptr=strstr(readback,check);
    if(!ptr)
    {//not found
     cnt=0;
     check[0]=255;
     check[1]=255;
     check[2]=1;
     check[3]=1;
     check[4]=0;
//
     do {
      strcpy(check2,outvalue(0));// just before cnt2++ BEFORE else. After check[4]=0
      ptr=strstr(check2,check);
      cnt++;
     } while((cnt<6)&&(!ptr));

     while((cnt<6)&&(!(ptr=strstr(outvalue(0),check)))) cnt++;
     if(cnt==6) giveback=-1; else giveback=0;
    //} else giveback=1;
    //
    } else if((inst1!=5)&&(inst1!=4))
    {
     if(check2[ptr-check2+3+i]==1) i++;
     if(check2[ptr-check2+4+i]==1) i++;
     giveback=check2[ptr-check2+5+i];
    } else giveback=1;

    cnt2++;
   }
  }
 } while((giveback==0)&&(cnt2<3));
 ////////
 if((giveback>0)&&( (inst1==4)||(inst1==5) ))
 {
  a1=(inst3&15);
  outvalue(a1);
  a2=(inst3&240)/16;
  outvalue(a2);
  a3=(inst3&3840)/256;
  outvalue(a3);
  a4=(inst3&61440)/4096;
  if((inst1==4)&&(inst2<3)) a4=a4+8*256;
  strcpy(readback,outvalue(a4));
  check[0]=255;
  check[1]=255;
  check[2]=inst1;
  a1=a1+a2*16;
  i=0;
  if(a1>1) check[3]=a1;else {check[3]=1;check[4]=1+a1;i++;}
  a3=a3+a4*16;
  if(a3>1) check[4+i]=a3;else {check[4+i]=1;check[5+i]=1+a3;i++;}
  check[5+i]=0;
  if(!(ptr=strstr(readback,check))) giveback=-1; else giveback=1;
 }
 return giveback;
}

char* outvalue(int outval)
{
```

```
unsigned char rdback[20];
int out=888;
int control=890;
int in=889;
int invalue;
int cnt;
int fla;
fla=(int)(outval/256);
outval=outval&255;

outportb(control, 3);
outportb(out,outval);
outportb(out,outval);
outportb(out,outval);
outportb(control, 6);
outportb(control, 6);
outportb(control, 6);
outportb(control, 3);
outportb(control, 3);
outportb(control, 3);

/// don't forget to establish reset of buffers after dsp-prgstart
 outportb(control,5);
 cnt=0;
 if(fla) delay(fla*4);
 do {
  cnt++;
  invalue=(inportb(in)&120) / 8;
 } while((!(invalue&2))&&(cnt<10000));

 outportb(control,3);

 cnt=0;
 if(invalue&2)
 {
  do {
   outportb(control,15);
   invalue=(inportb(in)&120) / 8;
   outportb(control,11);
   outportb(control,13);
   invalue=255-(invalue+(inportb(in)&120)*2);
   if(invalue>1) rdback[cnt]=invalue;
   else {rdback[cnt]=1;cnt++;rdback[cnt]=1+invalue;}
   cnt++;
   outportb(control,11);
  } while((inportb(in)&128)&&(cnt<19));// not yet empty!!
  if(cnt==19) {rdback[cnt]=1;cnt++;}
 }
 rdback[cnt]=0;
 outportb(control,3);
// read until empty
 return rdback;
}

int Do_it(int inst1, int inst2, int inst3)
{
 int z=0;
 int ar1;

 do {
  ar1=outinstruction(inst1,inst2,inst3);
  z++;
 } while((z<5)&&(ar1<1));       // Reset counter
```

```
 if(z==5) {beep(1000,300); printf("DSP does not respond!"); ar1=0;}
 return ar1;
}

int GetDSPvalue()
{
 int invalue;
 int control=890;
 int in=889;
 int out=888;
 int cnt=0;

outportb(control,5);
 do {
  cnt++;
  invalue=(inportb(in)&120) / 8;
 } while((!(invalue&2))&&(cnt<10000));
 if(cnt<10000)
 {
  outportb(control,3);
  outportb(control,15);
  invalue=(inportb(in)&120) / 8;
  outportb(control,11);
  outportb(control,13);
  invalue=255-(invalue+(inportb(in)&120)*2);
 } else invalue=-1;
 return invalue;
}
```

The Scanning Procedure:

```
void scan(int xnumber, int ynumber, int factor, int factorz, int voltagez, int referenceintensity)
{
  float size2;
long int summe[1024];
unsigned int readDSP[1024];// or use farmalloc and farcalloc
clrscr();
  ScanGraphicsInit(4);
int invalue;
    int control=890;
    int in=889;
    int out=888;
    int cnt=0;
    int direction;
    int x2;
    int choice;
    long int summ=0;

 for(int y=0; y<ymax; y++)
 {
  outportb(control,5);
  Do_it(9,5,0);
  for(int x=0; x<xmax-1; x++)  /// ;x<=xmax;
  {
   outportb(control,11);
   while((!bioskey(1))&&(!(inportb(in)&128)));
   invalue=(inportb(in)&120) / 8;
   outportb(control,13);
   readDSP[x]=255-(invalue+(inportb(in)&120)*2);
```

```
  /*
  if(readDSP[x]&3)
  {
  outportb(control,11);
  //while(!(inportb(in)&128));// if(bioskey(1)==283) {break;break;}// wait for busy <> 0 (==> not empty)
  invalue=(inportb(in)&120) / 8;
  outportb(control,13);
  readDSP[x]=255-(invalue+(inportb(in)&120)*2);
  }*/
  //while(!(inportb(in)&128));// if(bioskey(1)==283) {break;break;}// wait for busy <> 0 (==> not empty)
  outportb(control,11);
  invalue=(inportb(in)&120) / 8;
  outportb(control,13);
  readDSP[x]+=(long int)(255-(invalue+(inportb(in)&120)*2))*256;
  }
  if(bioskey(1)) if(bioskey(1)==7181) {x=xmax+1;getch();} else if (bioskey(1)==283) {x=xmax+1;y=ymax+1;getch();} else
getch();// wait for busy <> 0 (==> not empty)
  /*
  asm {
  xor di,di;
  mov dx,control;
  mov al,11;
  out dx,al;
  dec dx;
  }
 empty1:
  asm {
  in al,dx;
  test al,080h;
  jz empty1;
  shr al,3;
  and ax,0fh;
  test al,3h; //test if lo bits of lo byte where read (must be zero)
  jnz empty1;
  mov cx,ax;
  inc dx;
  mov al,13;
  out dx,al;
  dec dx;
  in al,dx;
  shl al,1;
  and al,0f0h;
  add cl,al;
  inc dx;
  mov ax,11;
  out dx,al;
  dec dx;
  }
 empty2:
  asm {
  in al,dx;
  test al,080h;
  jz empty2;
  shr al,3;
  and al,0fh;
  add ch,al;
  inc dx;
  mov al,13;
  out dx,al;
  dec dx;
  in al,dx;
  shl al,1;
  and al,0f0h;
  add ch,al;
```

```
 xor cx,0ffffh;
 mov word ptr [di+&readDSP[0]],cx;
 inc dx;
 mov ax,11;
 out dx,al;
 dec dx;

 inc di;
 cmp di,xmax;
 jle empty1;
 }
*/
beep(500,50);
//movedata(FP_SEG(readDSP), FP_OFF(readDSP),FP_SEG(fptr+y*(xmax+1)), FP_OFF(fptr+y*(xmax+1)),xmax+1);
if(y&1) direction=xmax-1; else direction=0;
summe[y]=0;
for(x=0;x<=xmax;x++) summe[y]+=readDSP[x];
summe[y]/=xmax+1;
if(y) summ=(int)((float)(summ*(y-1))/y+(float)summe[y]/y);else summ=summe[0];
for( x=0; x<xmax; x++)
{
 x2=abs(direction-x);
 //if(readDSP[x]<0) readDSP[x]=255;
 DisplayValue(x2,y,(readDSP[x])/256);
}
}
/*for(y=0;y<=ymax;y++)
{
 movedata(FP_SEG(fptr+y*(xmax+1)), FP_OFF(fptr+y*(xmax+1)),FP_SEG(readDSP), FP_OFF(readDSP),xmax+1);
 for(int x=0;x<=xmax;x++)
 {
 x2=abs(direction-x);
 DisplayValue(x2-1,y,(readDSP[x]-summ)/128);
 }
}*/
}
```

The Interface between User and PC (this part shows the basic usage of the defined sets of commands):

```
void menue()
{
 int choice,choice2=0;
 int voltage         = voltagemin;
 int stepsize        = 32;
 int voltagestep     = 8
 int referenceintensity;
 int calibratecontinue;
 long int amplitude=0xa0;//=0222h => Max ampl is SQR(1-(amp*frequ)^2)
 long int dotspercycle=0x190;//=78h ==> dotsperc.*ampl <ffffh.  The real frequency
                                        // is 465kHz/2/dotspercycle
 long int lrefa=0,dlrefa=0,lrefs=0,dlrefs=0,xpos=0,ypos=0,zpos=0,vbias=0;
 int k;
 long int novell;

 steps[0]=1;
 clrscr();
 menuewin();
 statuswin();
do {
 while(choice != ESC)
 {
  _setcursortype(_NOCURSOR);
  checkkey();
  choice = get_key();
  statuswin();
  switch(choice)
  {
   case UP  : {
                Do_it(6,5,0); //Reject Tip
                delay(3);
                Do_it(5,5,steps[0]); // Set # of steps
                last=Do_it(6,2,0);       // Walk forward
                clrscr();
                printf(" Walker moved %d steps forward.",steps[0]);
                menuewin();
                break;
               }

   case DOWN : {
                Do_it(6,5,0); //Reject Tip
                delay(3);
                Do_it(5,5,steps[0]); // Set # of steps
                last=Do_it(6,4,0);       // Walk forward
                clrscr();
                printf(" Walker moved %d steps backward.",steps[0]);
                menuewin();
                break;
               }
   case LEFT  : {
                Do_it(6,5,0); //Reject Tip
                delay(3);
                Do_it(5,5,steps[0]); // Set # of steps
                last=Do_it(6,1,0);       // Walk forward
                clrscr();
                printf(" Walker moved %d steps left.",steps[0]);
                menuewin();
                break;
               }
```

```
  case RIGHT : {
                        Do_it(6,5,steps[0]); //Reject Tip
                        delay(3);
                        Do_it(5,5,steps[0]); // Set # of steps
                        last=Do_it(6,3,0);      // Walk forward
                        clrscr();
                        printf(" Walker moved %d steps right.",steps[0]);
                        menuewin();
                        break;
                       }

  case F8   : {
                        processing();
                        menuewin();
                        break;
                       }

  case F1   : {
                        DSP_Setup_Menue();
                        menuewin();
                        statuswin();
                        break;
                       }
 }
 if((0==0)||(strcmp(DSP_Prg,"No Program")))
  switch(choice)
  {
   case F2   : {


window(1,1,80,1);
textcolor(BLACK);
textbackground(WHITE);
clrscr();
printf("    D S P - Parameter Setup,              (c) Michael Tasler 1995");
window(1,2,80,11);
textcolor(BLUE);
textbackground(CYAN);
clrscr();
printf(" F1 : Walker-Amplitude, -dots_per_cycle, and -#_of_steps (%xh, %xh, %xh)",amplitude,dotspercycle,steps[0]);
 printf("\n F2 : Iref, dIref for autoapproach (0='_', 8000h='-', FFFF='~')");
 printf("\n F3 : Irefs, dIrefs for scanning (0='_', 8000h='-', FFFF='~')");
 printf("\n F4 : z-position (1='_', 2000h='-', 3FFE='~')");
 printf("\n F5 : x-position ((NOT YET))");
 printf("\n F6 : y-position ((NOT YET))");
 printf("\n F7 : Bias Voltage V (0='-', -2000h='_', 1FFF='~')");
printf("\n F8 : Contract_z: move z slowly to 3fffh");// used to be spectroscopy
printf("\n F9 : Feedback-test");
printf("\n ESC : END ");
printtime();

 _setcursortype(_NOCURSOR);
 checkkey();
 choice2 = get_key();
 statuswin();
 switch(choice2)
 {
  case F1  : {
                        _setcursortype(_NORMALCURSOR);
                        printf("\n Quasi-Amplitude of walker cycles (old value: %xh): ",amplitude);
                        novell=accept_or_change(amplitude);
                        if(novell>=0) amplitude=novell;
                        if(amplitude) Do_it(4,1,amplitude);
```

```
                    printf("\n Dots per walker cycle (old value: %xh): ",dotspercycle);
                    novell=accept_or_change(dotspercycle);
                    if(novell>=0) dotspercycle=novell;
                    if(dotspercycle) Do_it(4,2,dotspercycle);

                    printf("\n Number of walker steps (old value: %xh): ",steps[0]);
                    novell=accept_or_change(steps[0]);
                    if(novell>=0) steps[0]=novell;
                    break;
                  }
case F2:   {
                    _setcursortype(_NORMALCURSOR);
                    printf("\n Iref (old value: %xh): ",Irefa);
                    novell=accept_or_change(Irefa);
                    if(novell>=0) Irefa=novell;
                    if(Irefa) Do_it(4,3,Irefa);

                    printf("\n dIref (old value: %xh): ",dIrefa);
                    novell=accept_or_change(dIrefa);
                    if(novell>=0) dIrefa=novell;
                    if(dIrefa) Do_it(4,4,dIrefa);

                    break;
                  }
case F3:   {
                    _setcursortype(_NORMALCURSOR);
                    printf("\n Irefs (old value: %xh): ",Irefs);
                    novell=accept_or_change(Irefs);
                    if(novell>=0) Irefs=novell;
                    if(Irefs) Do_it(4,5,Irefs);

                    printf("\n dIrefs (old value: %xh): ",dIrefs);
                    novell=accept_or_change(dIrefs);
                    if(novell>=0) dIrefs=novell;
                    if(dIrefs) Do_it(4,6,dIrefs);

                    break;
                  }
case F4:   {
                    _setcursortype(_NORMALCURSOR);
                    printf("\n zpos (old value: %xh): ",zpos);
                    novell=accept_or_change(zpos);
                    if(novell>=0) zpos=novell;
                    if(zpos) Do_it(4,15,zpos);
                    delay(100);
                    if(zpos) Do_it(4,15,zpos);// twice since buffered output
                   break;
                  }
case F7:   {
                    _setcursortype(_NORMALCURSOR);
                    printf("\n vbias (old value: %xh): ",vbias);
                    novell=accept_or_change(vbias);
                    if(novell>=0) vbias=novell;
                    if(vbias) Do_it(4,9,vbias);
                    delay(100);
                    if(vbias) Do_it(4,9,vbias);
                   break;
                  }
case F8: {Do_it(6,5,0);break;}
case F9: {Do_it(6,7,0);break;}

}
                    _setcursortype(_NOCURSOR);
                    clrscr();
```

```
                    menuewin();
                    statuswin();
                    break;
                   }

    case F3   : {
                   last=Do_it(7,7,0); // Do next step
                   menuewin();
                    break;
                   }
    case F4          : {
                    _setcursortype(_NORMALCURSOR);
                    printf(" This option provides the possibility of setting breakpoints.");
                    printf("\n The DSP stops right before the function according to the value of the");
                    printf("\n breakpoint is about to be executed.");

                    printf("\n\n General float chard: 1-2-3-1-2-3-... until tunneling. Then 5-13-5-13-...-6 / 7");

                    printf("\n\n 0 : <cancel>                    1 : contract z");
                    printf("\n 2 : Step 1 walker step forward      3 : Expand z while monitoring I(z)");
                    printf("\n 4 : <continuous mode>             5 : Scann x position");
                    printf("\n 6 : End of Image                  7 : End of Image or z is Out of Range");
                    printf("\n 13: Scan next Line or until End of Line");
                    printf("\n\n Your choice (ENTER will confirm the old value)? ");

                     novell=accept_or_change(until);
                     if(novell>=0) until=novell;
                     _setcursortype(_NOCURSOR);
                     if(until) {if(until!=4) last=Do_it(9,until,0); else last=Do_it(8,8,0);}
                     clrscr();
                     if(last>0) printf(" DSP is executing until function %d.",until);
                     menuewin();
                    break;
                   }

     case F5          : {
                    last=Do_it(0,0,0); // pause DSP
                    printf(" DSP is paused");
                    menuewin();
                    break;
                   }

     case F6  : {
                    printf("\n Auto Approch:");
                    Do_it(6,5,0);      // rejects tip and gets AR1 away from 5 (is the case if autoapproach was done
before);
                    delay(3);
                    Do_it(5,5,1);      // Reset counter
                    Do_it(9,5,0);      // Wait for end of autoapp.
                    menuewin();
                    break;
                   }
     case F7          : {
                   scanmenue(1,1);// (should also include Voltage ramps);
                   clrscr();
                   menuewin();
                   break;
                   }
     case F10  : {
                    Do_it(6,5,0);// contract z
                    delay(3);
                    Do_it(5,5,1000);
                    Do_it(6,4,0);// walk 1000 steps backward
                    printf("\n Scanner is removed and tip rejected.");
```

```
                    break;
                   }
   }
   //++v=read_voltage(channel)+AD_voltage_offset;
   //++PlotParameters(stepsize,voltage-voltagemin,voltagemax-voltagemin,v);
 }

 if(strcmp(project,"No Project"))
  {
   beep(1500,50);
   window(1,23,80,25);
   textcolor(YELLOW);
   textbackground(RED);
   clrscr();
   printf("\n          You have to close the DSP-Project before you exit! <F1,F5>");
  choice=0;
  }
 } while(strcmp(project,"No Project"));

 Do_it(6,5,0); // reject tube
 delay(3);
 printf("\nEND");
 voltagestep=voltagestep;
}
```

(Appendix E): The DSP Software:

This appendix contains the complete source code of the DSP communication software.

The Root Program:

```
******* fbm7 *******
.mmregs
.include "var01.h"
.include "dspinitn.h"
.include "main.h"

.include "scannc.h"
.include "cont_zmc.h"
.include "read_ipc.h"
.include "walkerc.h"
.include "pspinit.h"
.include "calc0f3n.h" ; runs walker in different direction
.end
```

The Include File "var01.h":

```
; VAR01 contains all variables
DASIx  .set   56h
DASIy  .set   53h
DASIz  .set   55h
DASIv  .set   54h
PSPx0  .set   52h

 .ds 0f00h
SIN_SW .word   0000h        ; 0 -> random noise
y      .word   0h
y1     .word   0a1h         ; frequency:  *;*;*; y1 multiplied
                            ; the "*;*;*;*;*;*"-sequence must be 10000h !!!!!

; a-b-c, where a*c just smaller then 10000h. b=2c-2
;possible combinations are  249h-de0h-70h=7h,4
;Next try: 276h-ce0h-68h=dh,3
;next: 222h-ceh-0fh,3: that's it

y2     .word   0h
y2l    .word   0h
y2h    .word   0h
y3     .word   0h
offi   .word   0h

ta     .word   0
;; max speed is b=10 and d=4;;15&10:super
a      .word   5
aa     .word   5
b      .word   10;... 10 was here togethet wit 4 below at d

bb     .word   2
c      .word   21
cc     .word   4
d      .word   6;4      ;5     ; a,b,c,d are the values which are going
                                ; to be written in aa,bb,cc,dd respectively.
dd     .word   1
```

```
lhi    .word   0ffffh
delta_I .word   07a00h
lrefhi .word   0a000h
dlref  .word   00100h
n03fefh .word   03fefh
n08000h .word   08000h
n02000h .word   02000h

.ds    01010h
coeff  .word   00h
ccnt   .word   0190h
offset .word   0h
richtung .word   0h
ystep  .word   0100h
xstep  .word   0100h

rings  .word   00h
;rings  .space  0ee0h
;ringe  .word   01fffh


        .ps 0806h
int3:   B      office4
tint:   B      walker
```

The Include File "dspinitn.h":

```
;dspinit.h : does all the register setups of the dsp and aic's

.ps 0a00h
.entry
START:  SETC   INTM          ; Disable interrupts

        ldp    #0
        SPLK   #020h,TCR      ; T
        SPLK   #02ah,PRD      ; for AIC master clock (max:20h & 2ah)

        OPL    #0834h,PMST
        LACC   #0
        SAMM   CWSR           ; Set software wait state to 0
        SAMM   PDWSR          ;

        ldp #0
        CLRC   OVM            ; OVM = 0
        SPM    0              ; PM = 0

        ldp #0
        ;call hier1
        ldp #0
        SPLK   #0ch,IMR       ; useing tint as interrupt
        lacl #04h ; AICinit-mode
        samm AR1
        call AICINIT
        setc intm
        call calc

        ldp #0
        lacl #04h
        samm IMR
        lamm IFR
        samm IFR
        lacl #08h
        samm 51h
```

```
lacl #0
samm PSPx0

ldp #0
splk #03fffh,dbmr
lar AR0,#03ff0h;10h; z-count
lar AR1,#8003h; read_I mode
lar AR2,#02000h; x-count
lar AR3,#02000h; y-count
lar AR4,#02000h; x-slope:02000h=incr,0fdffh=decr
lar AR5,#02000h; y-slope:02000h=incr,0fdffh=decr
lar AR6,#rings; walker-cycle
ldp #xstep
lar AR7,#xstep

lacc #03fffh;#0
ldp #offset
sacl richtung
sfr
sacl offset

ldp #offi
mar *,ar2
lamm ar2
xor ar4
samm DASIx
lamm ar3
xor ar5
samm DASIy

CLRC   INTM          ; enable
lamm 51h
bsar 8
sacl offi
lacc #01ff0h
samm DASIv
lamm ar0
samm DASIz

lacl #0
samm tdxr
samm PSPx0
clrc sxm
```

The Include File "main.h":

```
;; Main.h  : establishes comunication to PC

rpt 15
 lamm 50h
lacl #0
samm 0ah;  preset Status C to "stop before setpoint"
lacl #0
samm 09h; preset walker movement as stop point;
lacl #01
samm 1fh

;0ah=statusC (command); AR1 is recent status; 09h is StatusB (used for
; second transmission and for compare register: "run until AR1=StatusB")
;1fh is counter for walker steps


wait:
```

```
    lamm 5ah
    and #1,11
    bcnd notempty,neq
    lamm AR1;       Status
    sub #1,15;
    bcnd wait,lt;   Wait for current Job to be over (over if <0)

    Setc INTM
    lamm 0ah;       StatusC
    bcnd wait,eq;   =0? => stop
    sub #8;
    bcnd noncont,lt; StatusC<8? =>non-continous mode
    sub #1
    bcnd cont,neq ; JMP for not 9.  C=9 is "continue until ar1=StatusB"
     lamm AR1;+;
     xor #1,15;+;
     samm AR1;      Status
    and #0fh
    ldp #0;
    sub 09h; Compare to Status B, which is "run until AR1=StatusB"
    bcnd nc2,neq
     lamm AR1
     or #1,15
     samm AR1; Pause (Bit 15 of AR1 =1)
     lacl #0
     samm 0ah; Pause (C=0)
     b wait
cont:
     lamm AR1;+;
     xor #1,15;+;
     samm AR1;      Status
     b nc2
noncont:
    add #2; check for c=6
    bcnd nc2,neq
     lamm 1fh ; counter
     sub #1   ; decrease
     bcnd nc11,eq ; jmp out if it was 1 (keep it at value '1')
     samm 1fh ; save decreased counter
     call walk
     b nc3

nc11:
     lamm 09h
     and #1
     bcnd nc3,neq
     lacc #0111h
;;    samm DASIv
     b nc3

nc2: lacc #3ffffh
    ldp #offset
    sacl richtung
    sfr
    sacl offset
    ldp #0
nc3: lamm 51h
    clrc INTM
    b wait;         and loop


notempty:
     lamm 50h;      Read from PC
     and #0fh,0
```

```
      samm 0ah;       StatusC
       lacl #081h; 129 is my convention for "first instruction received"
       samm 50h
       lamm 0ah
       samm 50h
      lamm AR1
      or #1,15
      samm AR1 ;       Stops current Process

     call getnext
     samm 09h;       StatusB
     bcnd nzero,neq
     lamm 0ah;
     sub #5
     bcnd nzero,eq; only if c=5, then second transmission=0 permittet!
     lacl #0
     samm 0ah ; StatusA: if second transmition is zero, always stop the system

nzero: lamm 0ah ;StatusC
     bcnd send,eq; if c=0 (Pause)
     sub #07
     bcnd single,lt;;; needsd to set AR1
     bcnd conti,gt     ;; continuous-procedure for c>=8
     lamm AR1;+;
     xor #1,15;+;
     samm AR1;       Status
     samm 09h;       StatusB
     b send
conti:
     lamm AR1;+;
     and #7fffh;++;
     samm AR1;       Status
     b send
single: ; singlestep procedure for c<7
     add #2
     bcnd count,eq ; set counter for walker if c=5
     add #1
     bcnd setvalue,eq; set value #B if c=4
     ;;; from now on c=6 or c=1,2 or 3
      lamm 09h; load second transmission value
      sub #1; walk left
      bcnd walk_l,eq
      sub #1; walk forward
      bcnd walk_f,eq
      sub #1; walk right
      bcnd walk_r,eq
      sub #1; walk backward
      bcnd walk_b,eq
      sub #1; contract z
      bcnd reject,eq
      sub #1;; readI(expandz)
      bcnd expand,eq
      sub #1;; scanx
      bcnd send,neq
            lacc #01fffh
            samm DASIv
            lacl #5;+; scanx mode
            samm AR1; pause mode
            lar AR7,#xstep
            ldp #0
            splk #03fffh,dbmr
            clrc sxm
            lacl #0
            samm PSPx0
```

```
            mar *,ar2
    b send

reject: lamm ar0
            ldp #n03fefh
            sub n03fefh
            ldp #0
            bcnd send, geq

            lacl #02h
            samm 51h
            ldp #0
            SPLK   #020h,TCR        ; T
            SPLK   #03ah,PRD        ; for AIC master clock (max:20h & 2ah)
            ldp #lhi
            lacl #01h;+; contract-z-mode
            samm AR1; pause mode
            b send

count:  ;(c=5)
    lamm 09h
    samm 1fh
    call get12bit
    b send
setvalue:
    lacc #7eh
    samm 50h
    lamm 09h
    samm 50h
    sub #1
    sfl
    sfl
    sacb ; and store #B in ACCB
     call getnext
     samm 1fh
     call get12bit
      call getPC
      clrc C;+1
      adcb ;+1; PC points on this instruction
      add #0fh ;+1; this has to point on num0!!!!!!
      cala ;+1
    lacb  ;+1; ACCB is (#B-1)*4. B=1 and B=2 are for Walker=>ACCB=0 & 4
    sub #4;+1
    cc calc,leq;+2; re-calculate walker cycle
    lacl #1;+1
    samm 1fh;+1
    b send ;+2
getPC:  pop  ;+1
            push ;+1
    ret   ;+1

num1:
            ldp #y1
            lamm 1fh
            sacl y1
            ret
num2:
            ldp #ccnt
            lamm 1fh
            sacl ccnt
            ret
num3:
            ldp #lhi
            lamm 1fh
```

```
        sacl Ihi
        ret
num4:
        ldp #delta_I
        lamm 1fh
        sacl delta_I
        ret
num5:
        ldp #Irefhi
        lamm 1fh
        sacl Irefhi
        ret
num6:
        ldp #dIref
        lamm 1fh
        sacl dIref
        ret
num7:
        ldp #xstep
        lamm 1fh
        sacl xstep
        ret
num8:
        ldp #ystep
        lamm 1fh
        sacl ystep
        ret
num9:
        lamm 1fh
        nop
        samm DASIv;; no xor. Thus 0=0 and +=max and -=min
        ret
numa:
        lamm 1fh
        samm AR2;;;x
        samm DASIx;;; wrong, since xor is missing
        ret
numb:
        lamm 1fh
        samm AR3;;;;y
        samm DASIy  ;;; wrong, since xor is missing
        ret
numc:
        ldp #xstep
        lamm 1fh
        sacl xstep
        ret
numd:
        ldp #xstep
        lamm 1fh
        sacl xstep
        ret
nume:
        ldp #xstep
        lamm 1fh
        sacl xstep
        ret
numf:
        lamm 1fh
        samm AR0;;;z pos : BE CAREFULL: Z-STEP WON'T BE SLOWED DOWN!!
        ldp #n02000h
        xor n02000h
        samm DASIz
        samm DASIz
```

```
            samm DASIz
            samm DASIz
            ret


send:
                        ;(B and 128) will tell PC, whether B>0 or B<0
                        ;(C and 192) will tell PC, whether C <,=,>0
                        ;           =0 =>c=0, =192 => c<0, =64 =>c>0
    lacc #0ffh
    samm 50h
    samm 50h
    lamm 0ah;       StatusC
    samm 50h;
    bcnd null,eq
    sub #8
    bcnd null,eq
    add #1
    bcnd null,eq
     lamm 09h;      StatusB
     samm 50h
     lamm 0bh;      StatusD
     b notnull
null:  lamm ar0
      samm 50h
      bsar 8
notnull:
    samm 50h;
    lamm AR1;       Status
    samm 50h

    sub #1
    bcnd nn2,eq
    sub #2
    bcnd nn1,neq
     lamm ar0;; if next step = Read_i then check if z not already at zmin
     sub #11
     bcnd nn1,gt
nn0:   lamm ar1
      or #1,15
      samm ar1
nn1: lamm 51h
    clrc INTM
    b wait;         and loop

nn2: lamm ar0
    ldp #n03fefh
    sub n03fefh
    bcnd nn0,geq
    b nn1

walk_l: b walk_f
walk_r: b walk_b
walk_f: lacc #3ffffh
            ldp #offset
            sacl richtung
            sfr
            sacl offset
            b walk
walk_b: lacl #0
            ldp #offset
            sacl offset
            sacl richtung
```

```
walk:
        lacl #00h
        samm 51h
        lacl #0ch;#08;#0ch
        samm IMR
        lamm IFR
        samm IFR
        ldp #0
        SPLK   #020h,TCR        ; T
        SPLK   #02ah,PRD        ; for AIC master clock (max:20h & 2ah)
        lacl #02;;; lamm 09h
        samm AR1
        mar *,AR6
        lacl *
        samm DASIv
        clrc intm
        b send

getnext:
    lamm 5ah
    and #1,11
    bcnd getnext,eq;; "wait for next value" MUST be in here!!!
    lamm 50h;       Read from PC
    and #0fh,0
    ret

get12bit: ;(first 4 bits need to be in 1fh)
    call getnext
    samm 09h;       StatusB
    lamm 1fh
    ldp #0
    add 09h,4
    samm 1fh
    samm 09h

    call getnext
    samm 0bh;       StatusD
    lamm 1fh
    ldp #0
    add 0bh,8
    samm 1fh

    call getnext
    samm 0bh;       StatusD
    lamm 1fh
    ldp #0
    add 0bh,12
    samm 1fh
    bsar 8
    samm 0bh
    ret

expand:
        lacl #1
        samm 1fh
        lacl #04h
        samm IMR
        lamm IFR
        samm IFR
        lacl #08h
        samm 51h
        lacl #03h;+; rd-I-mode
        samm ar1;
        lamm ar0;; if next step = Read_i then check if z not already at zmin
```

```
            sub #11
            bcnd send,gt
            lamm ar1
            or #1,15
            samm ar1
      b send
```

```
; idea: PC writes to StatusB
;      Interupt-Procedures write to StatusD
;      StatusC will determine the mode: 0=Stop (wait for PC)
;                                       >=8 continous (StatusD --> Status)
;                                       <8 PC-driven (StatusB --> Status)
;                                            (0 (wait)--> StatusB)
; Status = AR1; StatusB,C,D are 09h,0ah,0bh
; Requirements for StatusB and D: bits 6 and 7 must be 0
```

The Include File "scannc.h":

```
;scan: scans over whole range
; fullfill all prerequisits before!!!!

; 01ch summ register

office4:;;1=contrz,2=walk,3=readl,4=setAIC,5=scanX,6=EndOfImage,
            ;;7=scanX-OutOfRange,5+8=scanY,7+8=scanY-OutOfRange
            lamm 51h
            lamm AR1
            and #08007h
            sub #1
            bcnd contract_z,eq
            sub #2
            bcnd read_l,eq
            ;and #0dh
            sub #2
            bcnd scan,eq
            sub #2
            bcnd scan,eq
            rete

scan:
  lamm PSPx0
  ;; make summ her: a*oldvalue+b new values (save it in 01ch)
  ldp #0
  xor #08000h
  sacb
  add 01ch,2
  add 01ch
  bsar 2
  samm 1ch
  lacb
  ldp #Irefhi
  sub Irefhi
  bcnd zdec,gt
  add dIref
  bcnd zinc,lt
scan2:
  lacl #0
  samm 08h;;flag(=1 => dec, =2 => inc)
```

```
  ldp #0
  splk #0,arcr
  bit ar1,0ch
  bcnd scanx,ntc
scany:
  mar *,ar3
  mar *+
  bit ar3,1
  mar *-
  xc 2,tc
   xpl ar3
   xpl ar5
  mar *+,ar7
  lamm ar3
  xor ar5
  samm DASIy
  cmpr 0
  mar *-,ar3
  xc 1, ntc
   rete
  lar ar1,#8005h
   ldp #xstep
   lacl xstep
   sfr ; since indx=2
   samm ar7
  mar *,ar2
  b send_z
scanx:
 mar *+,ar7
 lamm ar2
 xor ar4
 samm DASIx
 cmpr 0
 bcndd scanx1,ntc
  mar *-,ar2
  mar *+
   ldp #xstep
   lacl xstep
   samm ar7
   ldp #0
;warten:
;  lamm 5ah
;  and #1,8
;  bcnd warten,eq
  lamm ar0
  sfl
  sfl
  and #0fffch
  samm 50h
  bsar 8
  samm 50h
scanx1:
 bit ar2,1
 mar *-
 xc 1,ntc
  rete
 xpl ar2
 xpl ar4
 lar ar1,#800dh
   ldp #ystep
   lacl ystep
   sfr ; since indx = 2
   samm ar7
  mar *,ar3
```

```
send_z:
;warten2:
;  lamm 5ah
;  and #1,8
;  bcnd warten2,eq
  lamm ar0
  and #0fffch
  samm 50h
  bsar 8
  samm 50h
  ;;lamm ar0; the summ is 01ch
  ;;and #7fh
  ;;samm 50h
  ;;lamm ar0; the summ is 01ch
  ;;bsar 7
  ;;or #80h
  ;;samm 50h
  ;lamm 01
  ;and #07fh
  ;samm 50h
  ;lamm 01
  ;bsar 7
  ;or #80h
  ;samm 50h
  ;;lacl #0
  ;;samm 01ch
  ;samm 01
 rete

zdec:
            ;;lacl #0
            ;;samm PSPx0
            lamm 08h
            sub #2;  already inc?
            bcnd scan2,eq
            lacl #1 ; set dec mode
            samm 08h

            lamm ar0
            sub #1
            samm ar0
            xor #02000h
            samm DASIz
            lamm ar0
            sub #2
            bcnd zdec2,gt
            ; OUT OF RANGE
             lamm ar0
             add #1
OOR:    samm ar0
            rete

             lamm ar1
             or #02h
             samm ar1
             or #0f
             ldp #0
             sub 09h  ;; compare to Stutus B
             bcnd scan2,neq
             lamm 0ah ;statusc
             sub #09;; to check for "run until"
             bcnd scan2,neq;; stop only if "run until OOR"
             rete
zdec2:
```

```
            ;lamm ar1
            ;and #0dh ;=f-2
            ;samm ar1
            rete

zinc:
            lamm 08h
            sub #1
            bcnd scan2,eq
            lacl #2
            samm 08h
            ;;lacl #0
            ;;samm PSPx0
            lamm ar0
            add #1
            samm ar0
            xor #02000h
            samm DASlz
            lamm ar0
            sub #03ffeh,0
            bcnd zdec2,lt
            ; OUT OF RANGE
             lamm ar0
             sub #1
             ;b OOR
            samm ar0
            rete
```

The Include File "cont_zmc.h":

```
;cont_zmc:(m for minus)(c for controllable)  drives piezo in home position
; for th case that some trouble occures: I changed line 17 from 0c to 08
contract_z:
            lamm ar0
            add #10h;sub #20h     ;;;;;;20!!!
            samm ar0
      ldp #n02000h
            xor n02000h
            samm DASlz
            lacl #0
            samm PSPx0
            lamm ar0
            sub n03fefh; #11h
            bcnd conz1,lt;gt

      lacc #03fefh
      samm AR0

            lacl #00h
            samm 51h
            lacl #08h;0ch
            samm IMR
            lamm IFR
            samm IFR
            ;-;lamm AR1
            lacl #02h;+; walker-mode
            or #1,15
            samm AR1; pause mode
            ;-;lacl #02h; walker-mode
            ;-;samm 0bh; to StatusD
            ldp #0
            SPLK    #020h,TCR         ; T
            SPLK    #02ah,PRD         ; for AIC master clock (max:20h & 2ah)
            ldp #lhi
```

```
        reti
conz1:  rete
```

The Include File "read_ipc.h":

```
;read_I: expands piezo as long as I within the limits (= no tunneling)

read_I:
        lamm PSPx0
        ldp #Ihi
        xor n08000h
        subs Ihi
        bcnd rdI2,gt
        adds delta_I
        bcnd rdI2,lt

        lamm AR0        ; z-count
        sub #10h;add #10h
        samm AR0
rdI0:
        ldp #Ihi

        xor n02000h
        samm DASIz
        lamm AR0
        sub #11h;n03fefh
        bcnd rdI1,gt;lt
    lacl #10h
    samm AR0
        lacl #02h
        samm 51h
        ldp #0
        SPLK    #020h,TCR       ; T
        SPLK    #03ah,PRD       ; for AIC master clock (max:20h & 2ah)
        ldp #Ihi
        lacl #01h;+; contract-z-mode
        or #1,15
        samm AR1; pause mode
        reti
rdI1:   rete
rdI2:
        lacl #5;+; scanx mode
        or #1,15
        samm AR1; pause mode
        lar AR7,#xstep
        ldp #0
        splk #03ffffh,dbmr
        clrc sxm
        lacl #0
        samm PSPx0
        mar *,ar2
    reti
```

The Include File "walkerc.h":

```
;Walker: puts out one walker cycloid

walker:
    lamm ar1
    and #1,15;bsar 15
    retc neq
        mar *,AR6
```

```
        lacl *+
        ldp #richtung
        xor richtung
    add offset;;++
        samm DASIv
        lacl #0
        samm PSPx0
        cmpr 0
        bcnd walk1,NTC
        lamm 1fh
        sub #1
        samm 1fh
        bcnd walk1,neq
        lacl #1
        samm 1fh
        lacl #04h
        samm IMR
        lamm IFR
        samm IFR
        lacl #08h
        samm 51h
        lacl #03h;+; rd-I-mode
        or #1,15
        samm ar1; pause mode
        lacl #0
        samm PSPx0
        reti
walk1:  rete
```

The Include File "pspinit.h":

```
;PSPinit: sets PSP2 to values given in VAR01.prt
* aic initialization data
*
AICINIT:
home:
    lacl #80h
    samm 51h
    ldp #0
    SPLK   #0ffh,IFR

    ;----------------------
    SETC   SXM          ;
    clrc intm   ; enable interrupts
    lamm 51h

    lacl #0h
    samm 52h
    nop
    nop
    nop
    nop
    nop
    nop
    nop
    nop
    nop
    nop
    nop
    nop
    nop
    nop
    nop
    samm 52h
```

```
        idle
        ;-----------------------
        LDP    #a             ;
        LACl   a
        add    aa,8
        sacl   ta
        CALL   AIC_2ND
        ;-----------------------
        LACl   b
        add    bb,8
        sacl   ta
        CALL   AIC_2ND
        ;-----------------------
        LACl   c
        add    cc,8
        sacl   ta
        CALL   AIC_2ND
        ;-----------------------
        LACl   d
        add    dd,8
        sacl   ta
        CALL   AIC_2ND

        RET

AIC_2ND:

        clrc intm   ; enable interrupts
        lamm 51h
        lacl #03h
        samm 52h
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        lacl  ta
        samm 52h
        idle
       RET
```

The Include File "calc0f3n.h":

```
;calc0f073: calculates Walker-cycloid in "#0fh,03h"-steps

calc:
            ldp #0
            splk #rings,cbsr1
            ldp #ccnt
            lacc ccnt,1
            sub #2
            add #rings,0
            samm cber1
```

```
        samm arcr
        ldp #0
        mar *,AR6
        lar AR6,#rings
        splk #0eh,cbcr
        lacl #0
        ldp #coeff
        sacl coeff
        setc sxm

fgh:
        LDP   #SIN_SW
        bit   SIN_SW,15
        BCND  SUBTRC,TC
        ldp #coeff
        lacl coeff
        add #1
        sacl coeff
        ldp #ccnt
        sub ccnt;    ;*;*;*;*;*;* 20 for 800
        ldp #coeff
        bcnd REC2,LT
        ldp #SIN_SW
        lacl #15
        sacl SIN_SW
        b REC2
SUBTRC:
        ldp #coeff
        lacl coeff
        sub #1
        sacl coeff
        bcnd REC2,GT

        lacl #0
        ldp #SIN_SW
        sacl SIN_SW
        mar *,AR6
        lar AR6,#rings
        ret

REC2:
        clrc SXM
        mpy #0
        lacl #0
        ldp #y1
        Macd coeff,y1
        apac
        sfr         ;==>Max 15 Bit long y1

        ldp #y2
        sacl y2
        sqra y2
        lacl #0
        apac
        sach y2h
        sacl y2l
        lacc #8,15
        rpt #11
        sfl
        sub #1      ; to get something like 7fffh ...
        sub y2h,16
        sub y2l     ;  (1-(p/pmax)^2), WHERE 1==1*2^30
```

```
            call sqroot ;  (sqroot of ACC, where ACCHI = vorkommastellen und
            acclo=nachkommastellen)

            setc SXM
            neg
            add #1fffh,2
            and #0fffch
            sfr;+;;++
            sfr
      ;;--xor  #02h,0ch;; don't turn walker direction _/\_
      ;;--add #1,3
            add #1,2 ;+;;++

            sacl *+
        b fgh


sqroot:
            Bcnd loo4,EQ
            sacl y2l
            sach y2h
            ldp #y3
            lacc #1,15
            sacb
            CLRC SXM
loo1:   sacl y3
            sqra y3
            lacc y2h,16
            add y2l
            setc SXM
            neg
            apac
            clrc SXM
            Bcnd loo2,NC
            lacl y3
            sbb
            sacl y3
loo2:   lacb
            sfr
            Bcnd loo3,C
            sacb
            lacl y3
            adcb
            B loo1
loo3:   lacl y3
loo4:   ret
```

BIBLIOGRAPHY

1) R. M. Feenstra, Physical Review B **50**, 4561 (1994).

2) R. M. Feenstra et al., in Proceedings of the Conference on Physical Properties of Semiconductor Interfaces at Sub-Nanometer Scale, 1992, edited by H. W. M. Salemink (Kluwer, Dordrecht, to be published)

3) P. Horowitz and W. Hill, The Art of Electronics (Cambridge Univ. Press, New York, 1982) p. 288

4) Texas Instruments, TMS320C5x User's Guide, (1993)

5) Texas Instruments, TMS320C5x DSP Starter Kit User's Guide, (1994)

VITA

Michael Tasler was born in Freiburg im Breissgau, Germany on March 31, 1972, the son of Bernd-Rüdiger Tasler and Renate Tasler. After moving to Essen, Moers, Munich and Frankfurt he completed his work at the German high school Hanns-Seidel-Gymnasium at Hösbach, Germany in July 1991. He entered the University of Würzburg, Germany in November 1991. In September 1993 he received his "Vordiplom" in physics.
In September, 1994, he entered the Graduate School of The University of Texas at Austin.

Permanent Address: Schlossberg 2
63773 Goldbach
Germany

Tel.: 01149-602-154-0259
e-mail: MiT@PHYSICS.UTEXAS.EDU

Publications:

This work will be published at the "Texas Instruments 1995 DSP Solution Challenge" contest and will be submitted to Texas Instruments Incorporated before December 31, 1995.

This thesis was typed by the author.

Exhibit O

SCIENCE & ENGINEERING LIBRARY

Dictionary of Computing

Fourth Edition

RECEIVED
MAY 0 4 1997
SEAVER SCIENCE

Oxford New York Tokyo
OXFORD UNIVERSITY PRESS
1996

Oxford University Press, Walton Street, Oxford OX2 6DP
Oxford New York
Athens Auckland Bangkok Bogota Bombay
Buenos Aires Calcutta Cape Town Dar es Salaam
Delhi Florence Hong Kong Istanbul Karachi
Kuala Lumpur Madras Madrid Melbourne
Mexico City Nairobi Paris Singapore
Taipei Tokyo Toronto
and associated companies in
Berlin Ibadan

Oxford is a trade mark of Oxford University Press

Published in the United States
by Oxford University Press Inc., New York

© Market House Books, 1983, 1986, 1990, 1996

First published 1993
Reprinted 1983, 1984, 1985
Second Edition 1986
Third Edition 1990
Reprinted 1990, 1991 (twice), 1992
Fourth Edition 1996

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, without the prior permission in writing of Oxford University Press. Within the UK, exceptions are allowed in respect of any fair dealing for the purpose of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act, 1988, or in the case of reprographic reproduction in accordance with the terms of licences issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside those terms and in other countries should be sent to the Rights Department, Oxford University Press, at the address above.

This book is sold subject to the condition that it shall not, by way of trade or otherwise, be lent, re-sold, hired out, or otherwise circulated without the publisher's prior consent in any form of binding or cover other than that in which it is published and without a similar condition including this condition being imposed on the subsequent purchaser.

A catalogue record for this book is available from the British Library

Library of Congress Cataloging in Publication Data
(Data available)

ISBN 0–19–853855–3

Text prepared by Market House Books Ltd, Aylesbury
Printed in Great Britain by
Biddles Ltd, Guildford & King's Lynn

Preface

The world of comp
expand and to cross ne
awareness. Jargon gro
sion abounds as the f
domain of specialists
edge. In preparing
Computing, we have
for clear explanations
affect more and more a
terminology that acc
dictionary is aimed
and teachers of compu
be of value to profe
computer users.

The fourth edition c
tains nearly 6000 term
and has a compreher
system. Almost 1700 n
added and many of the
been extensively upc
recent advances in all a
especially in personal c
graphics, networking,
and computer security

..p, j = 0,1,...q, the
ed

$_{p,j+1}$ ... $_{p,j+1}(v)$

line basis function of

nary space-partitioning

L (balanced multiway
ree n (≥2). A *multiway
n in which the root node
nonterminal node other
ree k, where $n/2 \leq k \leq n$,
occurs at the same level.
by R. Beyer and E.
a structure provides an
trieval device.
a B-tree is a *B+ tree*,
a primary index to an
comprises two parts: a
ontaining an entry for
file, and a B-tree acting as
to the sequential index
e used in *VSAM.
a no nodes of degree one.

f *inkjet printer.

m c bubble memory.

ang election) A form
hanging that simply inter-
elements that are out of
e of passes through the file,
s exist. The method is not
*straight insertion.

ision of a *data file, serving
which records are located.
ally used in connection with
iques; and with indexing
ndex) where index entries
of records. In these circum-
or indexing will yield the
rt of the bucket; the location
rieval within the bucket will
searching.
ose electric charge is used as
nic *RAM. A fully charged
bucket, is equivalent to a
harged or *empty* bucket, is

equivalent to a logic 0. The charge may be passed through an array of capacitors and associated electronics, which together form a *bucket brigade*.

bucket sort An external sort in which the records to be sorted are grouped in some way, and each group stored in a distinct *bucket. Different buckets will probably be stored on different storage devices. If searching is to be performed on the data, then each bucket should contain records with the same hash value (*see* hashing). In this way all the records that might contain the required key may be fetched from the external memory at once.

buddy system A method of implementing a *memory management system. The available memory is partitioned into blocks whose sizes are always exact powers of two. A request for m bytes of memory is satisfied by allocating a block of size 2^{p+1} where

$$2^p < m \leq 2^{p+1}$$

If no block of this size is available then a larger block is subdivided, more than once if necessary, until a block of the required size is generated. When memory is freed it is combined with a free adjacent block (if one exists) to produce a larger block, always preserving the condition that block sizes are exact powers of two.

buffer 1. A temporary memory for data, normally used to accommodate the difference in the rate at which two devices can handle data during a transfer. The buffer may be built into a peripheral device, such as a printer or disk drive, or may be part of the system's main memory. *See* buffering.

2. A means of maintaining a short but varying length of magnetic tape between the reels and the *capstan and head area of a tape transport, in order that the acceleration of the tape at the reels need not be as great as that of the tape at the capstan. There are two principal types of buffer: *tension arm* and *vacuum column*. In the first, the tape passes over a series of rollers, alternate rollers being fixed in position and the rest being attached to a sprung pivoted arm, so that a variable length of tape is taken up in the resulting loops; in the second, the tape is drawn by a difference of pressure into a chamber whose width is just that of the tape. Vacuum column transports are more expensive and noisier but can handle higher tape speeds.

Streaming tape transports and many types of cartridge drives do not use buffers and are therefore limited to lower accelerations of the tape in the area of the head and (if there is one) capstan.

3. Any circuit or device that is put between two others to smooth changes in rate or level or allow asynchronous operation. For example, line *drivers can be used to isolate (or buffer) two sets of data lines.

buffering A programming technique used to compensate for the slow and possibly erratic rate at which a peripheral device produces or consumes data. If the device communicates directly with the program, the program is constrained to run in synchronism with the device; buffering allows program and device to operate independently. Consider a program sending output to a slow device. A memory area (the *buffer) is set aside for communication: the program places data in the buffer at its own rate, while the device takes data from the buffer at its own rate. Although the device may be slow, the program does not have to stop unless the buffer fills up; at the same time the device runs at full speed unless the buffer empties. A similar technique is used for input. *See also* double buffering.

buffer register A storage location or device for the temporary storage of information during the process of writing to or reading from main memory. It generally has a capacity equivalent to one byte or one word.

bug An error in a program or system. The word is usually used to mean a localized implementation error rather than, say, an error introduced at the requirements or system-design stage. *See also* debugging.

bug seeding *See* seeding.

bulk memory *Another name for* backing store.

bulletin board (BBS) A *teleconferencing system often run on a dedicated computer for use by enthusiasts who can connect their personal computers by means of *modems and telephone lines or network connections. The

Exhibit P

THE OXFORD ENGLISH DICTIONARY

SECOND EDITION

Prepared by

J. A. SIMPSON *and* E. S. C. WEINER

VOLUME IX

Look–Mouke

CLARENDON PRESS · OXFORD

1989

Oxford University Press, Walton Street, Oxford OX2 6DP
Oxford New York Toronto
Delhi Bombay Calcutta Madras Karachi
Petaling Jaya Singapore Hong Kong Tokyo
Nairobi Dar es Salaam Cape Town
Melbourne Auckland
and associated companies in
Berlin Ibadan

Oxford is a trade mark of Oxford University Press

© *Oxford University Press 1989*

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior permission of Oxford University Press

British Library Cataloguing in Publication Data
Oxford English dictionary.—2nd ed.
1. English language-Dictionaries
I. Simpson, J. A. (John Andrew), 1953-
II. Weiner, Edmund S. C., 1950-
423
ISBN 0-19-861221-4 (vol. IX)
ISBN 0-19-861186-2 (set)

Library of Congress Cataloging-in-Publication Data
The Oxford English dictionary.—2nd ed.
prepared by J. A. Simpson and E. S. C. Weiner
Bibliography: p.
ISBN 0-19-861221-4 (vol. IX)
ISBN 0-19-861186-2 (set)
1. English language—Dictionaries. I. Simpson, J. A.
II. Weiner, E. S. C. III. Oxford University Press.
PE1625.087 1989
423—dc19 88-5330

Data capture by ICC, Fort Washington, Pa.
Text-processing by Oxford University Press
Typesetting by Filmtype Services Ltd., Scarborough, N. Yorks.
Manufactured in the United States of America by
Rand McNally & Company, Taunton, Mass.

W. H. Ireland *Scribbleomania* 118 A Walpole for love of vertù far renown'd. 1830 Cunningham *Brit. Paint.* I. 236 This country at that period..exported swarms of men with the malady of vertu upon them. 1871 Smiles *Charact.* ix. (1876) 262 There [at Rome], the virtus or valour of the ancient Romans has characteristically degenerated into vertu, or a taste for knicknacks.

b. *man* (or *gentleman*) *of virtu*, a virtuoso.

1749 Fielding *Tom Jones* XIII. v, They..may be called men of wisdom and vertù (take heed you do not read virtue). 1787 *Gentl. Mag.* 1163/1 Being in company lately with several gentlemen of virtù, I found in their conversation frequent use of the word Taste in a sense I was unaccustomed to. 1811 Jeffrey in *Edin. Rev.* May 31 There are few things, about which men of *virtu* are more apt to rave, than the merits of the Grecian architecture.

c. *article, object, piece,* etc., *of virtu*, an article such as virtuosos are interested in; a curio, antique, or other product of the fine arts.

α. 1771 Goldsm. *Haunch of Venison* 8, I had thoughts in my chambers to place it in view, To be shown to my friends as a piece of virtù. 1825 T. Hook *Sayings* Ser. II. *Man of Many Fr.* (Colburn) 148 Soon were they doomed to withdraw their eyes from the innumerable bits of virtù which surrounded them. 1857 C. Sumner in S. Longfellow *Life Longf.* (1891) II. 343 Stirling's house is full of the choicest articles of virtù. 1879 S. C. Bartlett *Egypt to Pal.* iv. 74 An immense number of articles of virtu from Egypt are now scattered through the world.

β. 1815 J. Scott *Vis. Paris* (ed. 2) 116 The manufacture of some decoration, some piece of vertù, some elegant trifle. 1848 Thackeray *Van. Fair* xlix, Bareacres Castle was theirs,..with all its costly pictures, furniture, and articles of vertu. 1902 Snaith *Wayfarers* ii, Every object of *vertu* that I ever possessed.

2. A special branch of this study or interest.

1745-6 Mrs. Delany in *Life & Corr.* (1861) II. 429 Last Tuesday Mr. Bristowe, an uncle of Miss Dashwood's, dined here; he is a great virtuoso, understands all the *virtus* to perfection.

3. *collect.* Objects of art; curios.

Not always clearly distinguishable from sense 1.

1746 H. Walpole *Let. to G. Montagu* 17 June, My books, my *virtu*, and my other follies and amusements take up too much of my time to leave me much leisure to think of other people's affairs. 1768 *Let.* in J. H. Jesse *Selwyn & Contemp.* (1843) II. 308 My longing to see my own collection of *virtu* at Castle Howard is wonderful. 1773 W. Mason *Heroick Ep. Sir W. Chambers* 7 Whose orb collects, in one refulgent view, The scatter'd glories of Chinese Virtù. 1839 Barham *Ingol. Leg.* Ser. I. *Acc. New Play*, Some Vandal or Jew, With a taste for virtu, Has knock'd off his toes, to place, I suppose, In some Pickwick Museum. 1858 D. Costello *Millionaire of Mincing Lane* ii, Pictures, crockery, gimcracks of all kinds —what is generally known as *virtù*.

attrib. 1792 J. Wolcott (P. Pindar) *Ep. Sir W. Hamilton* Postscr. 4 What Britons, *knowing* in the *Virtù* trade, Soon as a grand discov'ry shall be made, Are near thee,..prepar'd to bite?

4. The distinctive qualities inherent in a thing or person.

1934 E. Pound *Eleven New Cantos* xxxvi. 27 Or say where it had birth What is its virtu and power. 1949 —— *Pisan Cantos* lxxiv. 11 In the light of light is the virtù. 1969 *Listener* 14 Aug. 214/3 Cromwell was shown in the same light—of a *de facto* sovereign come into power thanks to his *virtù*—by Clarendon. 1973 *Times Lit. Suppl.* 1 June 601/3 The pagan *virtù*, the 'civic humanism' of Machiavelli, had become the proud Christian freedom of the Huguenots.

virtual ('vɜːtjuːəl), *a.* (and *sb.*). Forms: 4, 7–8 vertual (7 -all); 5 *Sc.* wertual(e, -all; 5–7 virtuall (5 -alle), 6– virtual. [ad. med.L. *virtuālis*, f. L. *virtus* virtue, after L. *virtuōsus*. Hence also It. *virtuale*, Sp. and Pg. *virtual*, F. *virtuel*.]

1. a. Possessed of certain physical virtues or capacities; effective in respect of inherent natural qualities or powers; capable of exerting influence by means of such qualities. Now *rare.*

1398 Trevisa *Barth. De P.R.* XIX. viii. (Bodl. MS.), But vertual liȝt igadered in a litel place or in a pointe is cleped moche liȝt. 1477 Norton *Ord. Alch.* v. in Ashm. (1652) 62 But our chiefe Digesture for our intent, Is virtuall heate of the matter digerent. *a*1593 Marlowe *Hero & Leander* III. 89 So to all objects..his senses' flame Flowd from his parts with force so virtuall, It fir'd with sence things weere insensuall. 1626 Bacon *Sylva* §326 See if the Virtuall Heat of the Wine, or Strong Waters will not mature it. 1657 R. Ligon *Barbadoes* (1673) 106 Though the virtual beams of the Sun, give growth and life to all the Plants and Flowers it shines on. 1675 E. Wilson *Spadacrene Dunelmensis* Pref., Even ordinary water admits of a virtual mixture at least, as Experience evidenceth in Chalybeate waters. 1898 Meredith *Odes Fr. Hist.* 91 It was the foreign France the unruly feared,..Not virtual France, the France benevolent, The chivalrous.

b. Of herbs: Possessing specific virtues. *rare.*

1660 F. Brooke *Le Blanc's Trav.* 364 To Rivers they sacrifice the shels that come from them, to fountains fruits and vertual herbs. 1830 T. Aird *Captive of Fez* III, She knew..every virtual plant, and every sovereign flower Beneath the moon.

†**2.** Morally virtuous. *Obs.*

*c*1425 Wyntoun *Cron.* VII. 1218 His awyn oysse to lif wertual, May mirroure and ensample be Til alkyn statis. *Ibid.* VII. viii. 2206 Iohun of Salerne, prest cardynale, Commendit a lorde wertuale. 1607 Dekker *Wh. of Babylon* Wks. 1873 II. 216 You by your heauenly Influence change his vilenes Into a vertuall habit fit for vse.

†**3. a.** Capable of producing a certain effect or result; effective, potent, powerful. *Obs.*

1432–50 tr. *Higden* (Rolls) II. 177 For a man and the worlde be assimilate in iij. thynges, in dimension diametralle.., in disposicion naturelle, and in operacion virtualle. *Ibid.* 185. 1526 *Pilgr. Perf.* (Pynson) III. ix. 47b, That is called after Saynt Thomas virtuall attencyon which causeth a person in the begynnyng of his prayer to haue an actuall consideracion of the prayer or duety that he hath to do. 1619 W. Sclater *Exp. 1 Thess.* (1630) 37 So vertuall was the speech of Paul a Prisoner, in the heart of his Judge. 1640 Shirley *Arcadia* IV. iii, I meant it A draught for false Zelmane, it being virtual To increase affection. 1672 Josselyn *New Eng. Rarities* 12 The Loone is a Water Fowl, alike in shape to the Wobble, and as virtual for Aches. 1683 Moxon *Mech. Exerc., Printing* i, Dr. Dee..as a vertual Proof of his own Learned Plea, quotes two Authentique Authors.

b. *Mech.* (See quots.)

1815 J. Smith *Panorama Sci. & Art* II. 124 Whatever is the real length of the leg *b a* [of a siphon], the virtual or acting length when in use, only extends from *b* to the surface of the fluid. 1825 J. Nicholson *Oper. Mech.* 67 The velocity ..due to a head of 15 inches; and this we call the *virtual* or *effective head.*

4. a. That is so in essence or effect, although not formally or actually; admitting of being called by the name so far as the effect or result is concerned.

1654 Jer. Taylor *Real Pres.* 21 We affirm that Christ is really taken by faith,..they say he is taken by the mouth, and that the spiritual and the virtual taking him in virtue or effect is not sufficient, though done also. 1664 —— *Dissuas. Popery* ii. §8 But even this attention is not necessary that it should be actual, but it suffices to be virtual. 1697 Burghope *Disc. Relig. Assemb.* 166 We shall find it to amount to no less than a vertual renunciation of our baptism. 1734 Waterland *Diss. Exist. First Cause* 30 Every Proof *a priori* proceeds by Causes either real or virtual. 1769 Burke *Obs. Pres. St. Nat.* Wks. 1842 I. 112 One part of it could not be yielded..without a virtual surrender of all the rest. 1787 Bentham *Def. Usury* viii. 73 *Heading*, Virtual Usury allowed. 1820 Milner *Suppl. Mem. Eng. Cath.* 132 To prevent the virtual choice of a Catholic Bishop by an A-Catholic Ministry. 1844 H. H. Wilson *Brit. India* III. 211 He had reigned thirty-three years, during the first ten of which he was virtual sovereign of the greater part of Hindustan. 1883 A. Barratt *Phys. Metempiric* 157 The simplest conscious action involves actual or virtual thought.

b. *Virtual Church*, a council or similar body acting in the name of the whole church. Also *ellipt.* as *sb.*

1646 J. Maxwell *Burden of Issachar* 20 Whatsoever power,..the Catholike Church, or her virtuall and Representative, an œcumenical Councel, justly challengeth, this general Assembly vindicateth to it selfe. *Ibid.* 45 It was not consented to by the Church: that is, the Virtuall Church, the Generall Assemblie. 1654 Bramhall *Just Vind.* viii. (1661) 230 In all which..they understand..the virtual Church which is inuested with Ecclesiastical power, that is, the Pope with his Cardinals and Ministers.

c. *Optics.* Applied to the apparent focus or image resulting from the effect of reflection or refraction upon rays of light.

(*a*) 1704 J. Harris *Lex. Techn.* I, *Virtual Focus*, or *Point of Divergence* in a Concave Glass. 1728 Chambers *Cycl.*, *Point of Dispersion*, is that wherein the Rays begin to diverge; usually call'd the Virtual Focus. 1808 J. Webster *Nat. Philos.* 185 They issued from the virtual focus in the axis of the lens. 1831 Brewster *Optics* i. 11 The point A′, behind the mirror..is called their *virtual focus*, because they only tend to meet in that focus. 1874 *Lommel's Light* 90 The lenses of the second group have virtual foci.

(*b*) 1831 Brewster *Optics* ii. 18 In convex mirrors the image is always a virtual one formed behind the mirror. 1859 Parkinson *Optics* (1866) 130 A familiar instance of a *virtual* image is that formed by a common looking-glass of an object in front of it:—the image of an object under water is *virtual.* 1885 *Buck's Handbk. Med. Sci.* I. 39/1 If their direction, after the refraction, be prolonged backward, their prolongations meet to form a virtual image.

d. *Dynamics.* Of velocity or moment (see quot. 1867). *virtual displacement*, any notional, infinitesimal displacement in a mechanical system that is consistent with the constraints of the system; *virtual work*, the work done by a force making a virtual displacement.

1818 Barlow in *Encycl. Metrop.* (1845) III. 41/1 [The] principle..of virtual velocities..is now, by most foreign writers, made the foundation of the whole theory of statics. 1843 *Penny Cycl.* XXVI. 373/2 The name of the principle of virtual velocities..is very ill fitted to express the idea which is to be conveyed. [Full account follows.] 1867 Thomson & Tait *Nat. Phil.* I. i. §237 If the point of application of a force be displaced through a small space, the resolved part of the displacement in the direction of the force has been called its Virtual Velocity. *Ibid.*, The product of the force, into the virtual velocity of its point of application, has been called the Virtual Moment of the force. 1877 G. M. Minchin *Treat. Statics* iv. 61 The virtual work of a force is the product of the force and the projection along its direction of the virtual displacement of its point of application. 1897 A. E. H. Love *Theoret. Mech.* viii. 139 Principle of Virtual Work. The sum of the virtual works of all the forces on a system in equilibrium vanishes in every infinitesimal displacement. 1942 Synge & Griffith *Princ. Mech.* ii. 60 Although the chief merit of the principle of virtual work lies in the fact that it does not involve the reactions of constraints, nevertheless it can be used to find these reactions should they be required. 1981 R. R. Craig *Structural Dynamics* ii. 28 Use the principle of virtual displacements to derive the equation of motion of the idealized system shown below.

e. *Nucl. Physics.* Applied to an excited state of an atomic nucleus which has energy in excess of that needed for the emission of a particle but a lifetime sufficiently long for it to be regarded as a quasi-stationary state.

1931 *Proc. R. Soc.* A. CXXXIII. 228 According to the theory..the emission of α- particles by radio active nuclei is to be explained by the assumption that there exists in the nucleus a 'virtual' level of positive energy, which is occupied by an α- particle. 1955 I. Kaplan *Nucl. Physics* xvi. 368 Each excited state of the compound nucleus, whether bound or virtual, has a certain mean lifetime. 1963 W. E. Burcham *Nucl. Physics* ix. 372 All nuclear levels, except the ground state, can in principle emit radiation, leaving the nucleus in a less highly excited state, and virtual levels can in addition emit particles.

f. *Particle Physics.* Applied to particles and processes that cannot be directly detected and occur over very short intervals of time and space with correspondingly indefinite energy and momenta, which are not necessarily conserved within the time involved.

1949 *Physical Rev.* LXXV. 1305/2 These divergent terms must now be interpreted as renormalization or modification of the electric charge of the proton due to virtual mesons. 1961 W. S. C. Williams *Introd. Elementary Particles* xiii. 341 If the incident photon is 140 Mev and the positron is emitted at 90° with an energy of 100 Mev, then the four-momentum of this virtual electron is about 140 Mev/c. 1971 *Sci. Amer.* June 71/3 Although it may seem that virtual particles violate fundamental conservation laws, the violation is closely delimited to those areas where the uncertainty principle applies. 1973 L. J. Tassie *Physics Elementary Particles* viii. 15 The electron now consists of its 'bare' self together with all its virtual interactions with the electromagnetic field, corresponding to the electron emitting and re-absorbing virtual photons. 1973 *Sci. Amer.* Oct. 110/1 The scattering of the two electrons is described by saying that these particles exchange a virtual photon that transfers momentum from one particle to the other. 1979 D. R. Hofstadter *Gödel, Escher, Bach* (1980) v. 146 To understand how a real, physical electron propagates,..the physicist has to be able to take a sort of average of all the infinitely many different possible drawings which involve virtual particles.

g. *Computers.* Not physically existing as such but made by software to appear to do so from the point of view of the program or the user; *spec.* applied to memory that appears to be internal although most of it is external, transfer between the two being made automatically as required.

1959 *Proc. Eastern Joint Computer Conf.* XVI. 82/2 The sole function of the virtual memory is to increase machine speed. 1966 R. Adair et al. *IBM Cambridge Scientific Center Rep.* No. G320-2007 (*title*) A virtual machine system for the 360/40. 1966 *IBM Systems Jrnl.* V. 79 A virtual-storage computer (vsc) can decode addresses that are longer than those of its memory. The longer address is treated..as a virtual address that must be transformed to the actual, shorter memory address.... The virtual addressing of the word in external storage triggers a procedure that automatically brings the addressed word into memory. 1972 *Computer Jrnl.* XV. 199/2 Our system runs in a virtual machine, which is implemented by an interpreter. We can therefore easily add new instructions to our virtual hardware, merely by extending the interpreter. 1973 P. B. Hansen *Operating System Princ.* i. 3 An operating system makes a virtual machine available to each user.... The simultaneous presence of several users makes the virtual machines much slower than the physical machine. 1981 Pohl & Shaw *Nature of Computation* vi. 198 The Algolic language defines an Algolic virtual machine that may be implemented on a variety of computers. The Algolic machine could be constructed with the following software on a particular machine. 1982 G. Lee *From Hardware to Software* xxvi. 444 In a multi-programming system, several programs are being executed 'at once'.... Thus the operating system has to make available to each user a virtual store, of which he appears to be the sole user. 1983 *80 Microcomputing* Feb. 232/2 Virtual-memory systems have been prevalent in main-frames and large minicomputers for at least a decade. 1985 *Which Computer?* Apr. 54/1 No doubt this is a side effect of using the disc as a virtual memory.

h. Other collocations: *virtual cathode* (Electronics), a part of a space charge or electron beam where the potential is a minimum, so that electrons are repelled and positive ions attracted; *virtual height*, the height of an imaginary reflecting plane surface which in free space would give rise to the same travel time for reflected radio waves as an actual ionospheric layer; *virtual temperature* (Meteorol.) [tr. F. *température virtuelle* (Guldberg & Mohn *Études sur les Mouvements de l' Atmosphère* (1876) I. i. 6)], the temperature that dry air would have to have in order to have the same density as a given body of moist air when at the same pressure.

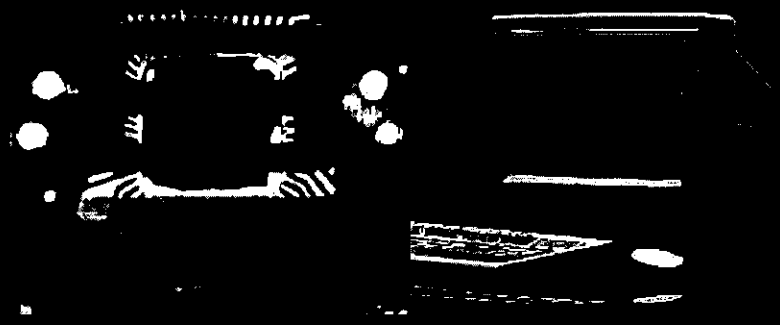
1937 *Virtual cathode [see suppressor 3]. 1964 *New Scientist* 1 Oct. 29/1 It was found that a virtual cathode could be obtained with a beam current of 3·5 milliamperes or more, and that its relaxation time was in fact inversely proportional to pressure. 1928 *Proc. IRE* XVI. 85 The heights as given in this paper are *virtual heights. They are calculated on the assumption that ordinary reflection takes place and that the layer is parallel to the earth's surface. 1967 [see ionogram 1]. 1975 D. G. Fink *Electronics Engineers' Handbk.* XVIII. 107 The reflection process for plane ionosphere is equivalent to mirror-type reflection at a height equal to the virtual height *h′* of reflection of the equivalent vertical frequency. 1910 C. Abbe tr. Guldberg & Mohn in *Smithsonian Misc. Collections* LI. No. 4. 124 We call the quantity *T* the *virtual temperature; for dry air the virtual temperature is the same as the absolute temperature. 1957 G. E. Hutchinson *Treat. Limnol.* I. vii. 468 *C* represents the virtual temperature of isothermal circulation prior to the development of stratification in the early summer. 1979 L. J. Battan *Fund. Meteorol.* v. 83 The effects of humidity can be taken into account by employing a quantity called the virtual temperature.

virtualism ('vɜːtjuːəlɪz(ə)m). [f. prec. + -ism.] The Calvinistic doctrine of Christ's virtual presence in the Eucharist.

Exhibit Q

DICTIONARY OF PERSONAL COMPUTING AND THE INTERNET





FD

FITZROY DEARBORN PUBLISHERS

nce

Case 1:07-mc-00493-RMC Document 188-18 Filed 07/23/2008 Page 3 of 4

DICTIONARY OF PERSONAL COMPUTING AND THE INTERNET

S.M.H. Collin

APR 15 1998



FITZROY DEARBORN PUBLISHERS
CHICAGO • LONDON

installed, Video clips can be played back using the Windows Media Player utility; sequences can be edited using the supplied VidEdit utility and video recorded with the VidCap utility; the quality of the video playback depends on the performance of the PC hardware, and the size of the playback window. With a window size of 160x120 pixels, any 486 or higher processor can display flicker-free video; for full-screen video playback at 640x480 pixels only a Pentium-based PC can display smooth motion

video game
game played on a computer that can be either an adventure game where you have to explore an electronic world or an arcade game in which you have to outwit or shoot lots of baddies!

video graphics array
see
VGA

video graphics card or overlay card
expansion card that fits into an expansion slot inside your PC and that allows a computer to display both generated text and graphics and moving video images from an external camera or VCR

video memory or video RAM (VRAM)
section of memory fitted on a video adapter that is used as a temporary store for image data sent from the computer's main memory or to store an image as it is built up and before it is displayed on the screen

VIM
VENDOR-INDEPENDENT MESSAGING
set of standards developed by IBM, Borland, Novell and Apple, that provides a way of sending electronic mail messages between applications
compare with
MAPI

virtual
something that does not actually exist, except in an imaginary form in a computer

virtual address
address that refers to a location in vitual memory

virtual desktop or screen
area that is bigger than the physical limits of the monitor, and which can contain text, images, windows, etc; the monitor acts as a window onto this area and a user can scroll around to view a different parts of the virtual screen; often used to help organise a window that has lots of icons

virtual image
complete image stored in memory rather than the part of it that is displayed

virtual memory
large, imaginary, main memory made available to an operating system by storing unused parts of the virtual memory on disk and then transferring these pages into

available main memory as and when they are required

virtual reality (VR)
simulation of a real-life scene or environment by a computer with which you can interact and explore

virtual reality modeling language
see
VRML

virus
software that is designed to corrupt your data files and copy itself automatically from one computer to another, usually by attaching itself to a normal application file; if you download a lot of programs from the internet, you should carry out a regular check with a virus checker that your computer has not been infected by a virus

virus checker
software that is used to try and detect and remove unwanted virus programs from the hard disk of your computer

Visual Basic™
programming tool developed by Microsoft, that allows users to create Windows applications very easily. Visual Basic is a rapid application development (RAD) tool that allows users with little programming knowledge to create a complex Windows application - although the finished applications are not as fast as programs written in C

Visual Basic Script™
see
VBSCRIPT

Visual Basic for Applications™ (VBA)
see
VBA

VL-bus or VL local bus
see
VESA

voice
i) (in MIDI) another name for a note or sound effect (such as a whistle); instruments that are multi-voice can play several notes at the same time.
ii) sound of a person speaking

voice mail
sophisticated telephone answering machine normally used in large companies, but also a feature of some new modems

volatile
electronic memory component that does not retain information when the electrical power is switched off

volume
convenient name used to identify a particular hard disk, CD-ROM drive or tape unit