January 1987

NEC v. Intel: Will Hardware Be Drawn into the Black Hole of Copyright Editors'

Robert C. Hinckley

Follow this and additional works at: http://digitalcommons.law.scu.edu/chtlj

Part of the Law Commons

Recommended Citation
Available at: http://digitalcommons.law.scu.edu/chtlj/vol3/iss1/2
NEC V. INTEL: WILL HARDWARE BE DRAWN INTO THE BLACK HOLE OF COPYRIGHT?

Robert C. Hinckley*

I. STATUS OF THE CASE

In this case, filed in the Northern District of California in December, 1984, NEC seeks, in the alternative, a declaration: (i) that the microcode portion of the Intel 8086/8088 microprocessors (the "Intel Microcode") is not, as claimed by Intel, subject to copyright protection; (ii) that Intel has forfeited its alleged copyrights because millions of copies of the Intel Microcode have been distributed to the public, under Intel's authority, without copyright notice; and (iii) that the microcode used by NEC in its V-Series microprocessors (the "NEC Microcode") does not infringe any valid and enforceable Intel copyright.¹

After a year and a half of intensive discovery and numerous motions, the case was tried in May, June and July, 1986 before Judge William A. Ingram without a jury. On September 22, 1986, Judge Ingram issued Partial Findings of Fact and Conclusions of Law² which held that (i) the Intel Microcode constituted copyrightable subject matter and (ii) the Intel copyrights had not been forfeited.

Prior to the announcement of the Partial Findings, NEC had informed the Court of the results of a just completed "clean room" project.³ The new microcode developed in the clean room sup-

---

¹ NEC also contends that it is licensed by Intel to use the Intel Microcode in the V-Series microprocessors, that Intel's alleged copyrights are unenforceable due to Intel's misuse of those copyrights, and that Intel has unfairly competed with NEC. Those contentions are not discussed in this article.

² 645 F. Supp. 590 (N.D. Cal. 1986).

³ See infra notes 180 to 187 and accompanying text. In a copyright infringement case, because direct evidence of copying is rarely available, a prima facie case of copying is typically proved circumstantially by showing (i) that the alleged infringer had access to the
ported earlier NEC evidence that the similarities between the Intel and NEC Microcodes urged by Intel as "evidence of copying"4 were in fact the result of constraints5 imposed on the designer of the NEC Microcode by similarities between the products' respective microarchitectures6 (for which NEC was licensed) and by implementation of the same, concededly unprotected, macroinstruction set.7 When Judge Ingram issued his Partial Findings with respect to copyrightability and forfeiture, he also ordered the testimony reopened so that NEC might introduce additional evidence regarding the results of that clean room development.

Shortly after the Partial Findings were received, NEC for the first time became aware of facts which appear to have required Judge Ingram's disqualification from the inception of the case. Prior to and during the course of the trial, a small, private investment club of which Judge Ingram was a member and officer held stock in Intel Corporation.8

When this was brought to Judge Ingram's attention by a letter protected work and (ii) that there is substantial similarity of protected expression between the allegedly infringing work and the copyrighted work. Sid & Marty Krofft Television Productions, Inc. v. McDonald's Corp., 562 F.2d 1157 (9th Cir. 1977). If it is shown that the defendant had no access of any kind, there can be no infringement even if the two works are identical. Mazer v. Stein, 347 U.S. 201, 217-218 (1954); Selle v. Gibb, 567 F. Supp. 1173, 1182-83 (N.D. Ill. 1983). The "clean room" approach eliminates possible access by providing to the persons involved in development of a competitive work only controlled information which, while describing the idea of the work to be developed, has been carefully screened, generally by counsel, to eliminate all expression of that idea which is the subject of the copyright that is to be avoided. Davidson, Common Law, Uncommon Software, 47 U. Pitt. L. Rev. 1037, 1096-97 (1986).

4. See infra notes 121 to 146 and accompanying text.

5. See infra notes 147 to 164 and accompanying text.

6. Microarchitecture refers to the design of the microprocessor as "seen" by a microcode designer. It includes storage for data ("registers") and pathways for moving data ("busses"), together with rules governing their use.

7. "Instruction set" or "macroinstruction set" means the collection of commands that a computer can perform. "Microinstruction set" refers to the set of possible defined bit patterns available to form a line of microcode.

8. 28 U.S.C.A. § 455 (West Supp. 1986) provides in relevant part:

(a) Any . . . judge . . . shall disqualify himself in any proceeding in which his impartiality might reasonably be questioned.

(b) He shall also disqualify himself in the following circumstances . . . .

* * * * *

(4) He knows that he . . . has a financial interest . . . in a party to the proceeding . . .

* * * * *

(c) A judge should inform himself about his personal and fiduciary financial interests, and make a reasonable effort to inform himself about the personal financial interests of his spouse and minor children residing in his household.
from special counsel retained by NEC, Judge Ingram requested that a formal recusal motion be made and stayed further proceedings. The recusal motion was referred to another judge for resolution, and as of the date of this article, that motion remains under submission. If Judge Ingram is ultimately found to have been disqualified, one further issue will be the effect such disqualification has upon his prior rulings, including the Partial Findings of Fact and Conclusions of Law.

II. HISTORICAL PERSPECTIVE

At stake in this litigation is whether Intel can use the copyright law to cripple a competitor's efforts to enter the marketplace with compatible products which are faster and better: the V-Series microprocessors. In order to appreciate the NEC/Intel litigation it is useful to understand the historical framework within which the case has been fought.

Until recently, the two companies had enjoyed a cooperative business relationship. In 1976, following extensive negotiations, NEC and Intel entered into a broad Patent Cross-License Agreement in which each company licensed the other under all existing patents and patents for which applications would be filed during the following 10 years, for the lives of those patents.\(^9\) The Patent-Cross License Agreement did not provide for any exchange of technology but committed the companies to hold periodic meetings, alternating at the home of each, so that technical matters of mutual interest

---

(d) For the purposes of this section the following words or phrases shall have the meaning indicated:

(1) "proceeding" includes pretrial, appellate review, or other stages of litigation;

   * * * * *

(4) "financial interest" means ownership of a legal or equitable interest, however small . . . except that . . .

(i) Ownership in a mutual or common investment fund that holds securities is not a "financial interest" in such securities unless the judge participates in the management of the fund . . .

   * * * * *

(e) No . . . judge . . . shall accept from the parties to the proceeding a waiver of any ground for disqualification enumerated in subsection (b). Where the ground for disqualification arises only under subsection (a), waiver may be accepted provided it is preceded by a full disclosure on the record of the basis for disqualification.

9. NEC Ex. 1.
could be discussed as well as the possibility of technology exchanges.

In early 1978, two years after executing the Patent Cross-License Agreement, Intel announced the introduction into the market of its 8086 microprocessor, hastily developed to compete with the highly successful Zilog Z80 microprocessor. To help achieve market acceptance, Intel encouraged NEC to become a second source for the 8086.10 Reliable second sources were critically needed by Intel at this time to give credibility to its product.11

In addition to developing and distributing its own original products, NEC did become an 8086 second source. It did so through a two-year reverse-engineering12 effort without assistance from Intel, but with Intel's knowledge and blessing. In fact, NEC sought and obtained Intel's prior approval of the NEC press release announcing the NEC microprocessor, known as the µPD8086.13 Intel used reliable second sources, including NEC, as a marketing tool with potential customers. As a result, the 8086 achieved a tremendous success and quickly became an industry standard microprocessor.

In developing the µPD8086, NEC believed that it was entirely free to duplicate the Intel Microcode since Intel had encouraged

---

11. Users of microprocessors, such as manufacturers of computers, automobiles and other products, generally insist on having alternative sources of supply before they will commit to incorporating a particular device into their products. This insistence is due to the dual concerns that supply not be interrupted and that there be price competition. Therefore, for a newly introduced microprocessor to achieve market success, second sources are essential.

The goal of a second source is to make a product which is fungible with the original product. For such a product to be truly fungible, it must be compatible on all levels with the original product. There are several levels of compatibility.

The lowest level of compatibility is referred to as “functional” or “instruction set” compatibility; that is, the products execute the same instructions. Functional compatibility implies, but does not always guarantee, “software” compatibility; that is, while some software programs intended to run on one device will operate correctly on the other, different programs, because of timing or other differences, may not. The next level is known as “pin” compatibility; that is, one product may be plugged into a socket intended for the other product and function as a replacement for the intended product, despite having different electrical characteristics. The highest level is known as “test” or “electrical” compatibility; that is, the two products are electrically identical. The goal of a second source is to be compatible on all levels with the original product.

12. Reverse engineering is a standard technique in which a company studies a competitor's product, and uses the knowledge gained to design a functionally similar competing product. Reverse engineering was acknowledged as a legitimate industrial method, different from copying, in the Semiconductor Chip Protection Act. See 17 U.S.C.A. § 906(a) (West Supp. 1986).
13. NEC Ex. 31, 32.
NEC to develop a second source product and because the Intel Microcode was part of the microprocessor's hardware and thus, if legally protectable at all, could only be protected by Intel's patents under which NEC was licensed. Accordingly, the NEC μPD8086 microcode was virtually identical to the Intel Microcode.\textsuperscript{14} Significantly, even though Intel had obtained an NEC μPD8086 in March, 1981, shortly after it was introduced, and had examined the inside of the chip, at no time before August 1982 did Intel raise with NEC any copyright issue concerning the microcode in the μPD8086.

Of course, some of the advantages which cause customers to insist on the existence of a second source for a microprocessor, such as price competition, are disadvantages from the perspective of the primary source, especially if the competitor is cost effective and technologically equivalent. Accordingly, in August, 1982, after the Intel 8086 was safely established as an industry standard and Intel's need for NEC as a second source was no longer critical (but seventeen months after Intel had first examined a μPD8086), Intel for the first time notified NEC that its μPD8086 infringed Intel's alleged copyright in the Intel Microcode.

NEC was shocked at the sudden turnaround in Intel's attitude regarding the μPD8086, especially since Intel had previously approved and encouraged NEC's development of that product. But NEC felt that, since its product was on the market and it had made commitments to customers, it was best to settle with Intel. In February, 1983, the two companies entered into a copyright license agreement.

NEC was determined, however, that it would not be put in such a position again. Even before the copyright license was executed, NEC, motivated in part by Intel's copyright claim, canceled its development plans for a low-power, CMOS\textsuperscript{15} version of the μPD8086 which would also have embodied the Intel Microcode. Instead, an NEC team was instructed to develop a new and improved family of proprietary CMOS microprocessors. These new

\textsuperscript{14} For the NEC μPD8086 to be a true second source product of the Intel 8086, i.e., compatible on all levels, it had to contain virtually the same microcode as the Intel 8086. In contrast, the V20/V30 are both pin compatible and instruction set compatible, but not test compatible with the 8086/8088. That is why the V20, for example, can be used to replace the 8088 in most personal computers, and will run most software written for the 8088, yet cannot be called a "second source" of the 8088. See infra note 16.

\textsuperscript{15} "CMOS" refers to a type of semiconductor device known for its low power consumption. The Intel 8086/8088 were NMOS devices which required significantly greater power to operate.
devices, called the V-Series,\(^\text{16}\) were to be upwardly compatible with the Intel 8086/8088; that is, they were to be capable of running the Intel macroinstruction set, plus many powerful new macroinstructions which the Intel 8086/8088 could not run. The V-Series were also to be faster and use significantly less power than Intel’s 8086/8088 microprocessors. NEC disclosed to Intel its plans for the V-Series within weeks after the development effort began.

To achieve higher performance, the NEC team chose a dual bus microprocessor design.\(^\text{17}\) While some of the basic V-Series microarchitecture was quite similar to that of the 8086 (for which NEC was licensed under the Patent Cross-License Agreement), the new V-Series microprocessors were nonetheless physically different from both the Intel 8086 and the NEC μPD8086 in many respects and, as a result, could not use the Intel Microcode. Physical differences such as the dual bus, hardware implementation of certain arithmetic functions, and a different microcode format resulted in NEC Microcode which, when laid beside the Intel Microcode, exhibits no similarity.\(^\text{18}\) At trial, the only way Intel was able to even suggest similarity between the Intel and NEC Microcodes was to eliminate, by numerous “translations”, the differences caused by the different physical structures of the respective devices.\(^\text{19}\)

When samples of NEC’s final prototype V20 and V30 microprocessors were completed and field testing was underway, NEC presented the V-Series to several selected audiences, including Intel. NEC hoped that Intel would become a second source for the V-Series as NEC had done for the Intel 8086/8088. During this period NEC furnished Intel with engineering samples of the new NEC devices, which of course included the NEC Microcode, so that Intel

---

\(^{16}\) The mainstays in the NEC V-Series Family are the V20, V30, V40, V50, V60 and V70 microprocessors. The family contains other microprocessors and complex peripheral devices. The V20 and V30 are upwardly software and pin compatible, but not test compatible, \((\text{see supra, note 11})\) with the 8088 and 8086 respectively. The V20 and V30 execute the 8086/8088 macroinstructions as well as those of the 80186 microprocessor, emulate the 8080 microprocessor \((\text{see infra, note 106})\) and have additional unique V-Series macroinstructions. The V40 and V50 microprocessors are the V20 and V30 microprocessors with additional peripheral functions contained on each chip. The V40 and V50, while upwardly software compatible, cannot be used as a direct replacement for any Intel product. The V60 and V70 are 32-bit microprocessors which maintain the upward software compatibility with the V-Series. The microcode in the V60 and V70 microprocessors was not involved in the suit.

\(^{17}\) The Intel 8086/8088 possesses a single bus, or path for transferring data, so only one transfer can occur at a time. The V20/V30 has two paths, so that two transfers can occur simultaneously.

\(^{18}\) See infra notes 101 to 106 and accompanying text.

\(^{19}\) NEC disputes the validity of Intel’s “translations.” See infra notes 106 to 110 and accompanying text.
not only could evaluate the desirability of becoming a second source but also could satisfy itself that the NEC Microcode was non-infringing.

A number of months passed during which NEC attempted to obtain a commitment from Intel to become a second source for the V-Series and/or an acknowledgment of NEC’s independent development of its new products. 20 In response, NEC received only excuses and promises that Intel’s analysis would soon be complete, but no answers. 21 Meanwhile, during this critical time in the marketing of the new NEC device, NEC became aware that questions concerning the legality of NEC’s new products were being raised among potential customers by unknown sources. Stories to the same effect started to appear in newspapers and the trade press. The negative impact of these rumors on NEC’s potential customers mandated the removal of this cloud as quickly as possible and NEC accordingly filed this action for declaratory relief.

III. THE LEGAL ISSUES

A. Copyrightability of the Intel Microcode

It is undisputed that computer hardware is not copyrightable, but only patentable. 22 It is now also settled that computer software is copyrightable. 23 The computer industry has long recognized, however, that microcode fits neatly into neither category; it is neither quite hardware nor quite software, but occupies an intermediate position which computer engineers have come to refer to as “firmware.” 24 Therefore, the issue of first impression in this case is

20. As was subsequently shown by internal Intel documents, it was quickly apparent to Intel that the NEC V-Series significantly outperformed the Intel products. NEC Ex. 208, 209. Intel was unprepared for such competition, and apparently would not have the capability to make its own low power CMOS 8086/8088 until over a year after the introduction of the V-Series. Tr. XVII 2486:3-2487:24. In the meantime, Intel was forced to purchase CMOS parts from competitors and package them under the Intel name. Tr. VII 1112:8-1113:6; 1116:19-1117:15; Intel Ex. JD.

21. Intel did finally, in late 1984, inform NEC that the results of its initial analysis of the NEC Microcode were “inconclusive.” Tr. VI 902:14-22.


24. Intel’s expert, Dr. David A. Patterson, published an article in Scientific American in which he classified microcode as “firmware, thereby signifying that intermediate status between hardware and software.” Patterson, Microprogramming, 248 Sc. Am. 50 (March 1983). NEC Ex. 449.
whether the proper legal protection to be afforded firmware such as the Intel Microcode is via the patent or copyright laws.

Reduced to its simplest terms, the question of whether the Intel Microcode is copyrightable can be stated this way: was the creator of the Intel Microcode designing a computer or programming a computer?\(^{25}\) If the former, then patent, not copyright, is the appropriate means of protection.\(^{26}\) The line between hardware and software has to be drawn someplace. On which side of that line the Intel Microcode falls must be determined according to the policies underlying our intellectual property laws.

The syllogism put forward by proponents of the copyrightability of the Intel Microcode is that all computer programs are copyrightable;\(^{27}\) the Intel Microcode is a computer program; therefore, the Intel Microcode is copyrightable.\(^{28}\) However, this simplis-

---

25. Unfortunately the term “microcode” has been and is used even by technical people to refer to a variety of structures and approaches. For example, there are specially designed computers which are programmable on the micro level. Programs written for those computers, primarily laboratory or university learning tools, are also called microcode. The programmer selectable sequences of microinstruction created to instruct those machines were not involved in NEC v. Intel. The issue at hand was whether the Intel Microcode, developed as an integral part of the computer design process, is more appropriately protected by copyright or patent. Throughout the remainder of this article, the term “microcode” refers only to the Intel Microcode.

26. It has already been established that firmware may constitute a part of a patentable invention. In re Bradley and Franklin, 600 F.2d 807 (C.C.P.A. 1979), aff’d by an equally divided court, 450 U.S. 381 (1981).


28. Intel also argued that its Microcode was copyrightable on the ground that it was a “literary work.” 17 U.S.C.A. § 102(a) (1977) sets forth the general requirements for copyrightability as well as categories of copyrightable subject matter:

Copyright protection subsists, in accordance with this title, in original works of authorship fixed in any tangible medium of expression, now known or later developed, from which they can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device. Works of authorship include the following categories:

(1) literary works;
(2) musical works, including any accompanying words;
(3) dramatic works, including any accompanying music;
(4) pantomimes and choreographic works;
(5) pictorial, graphic, and sculptural works;
(6) motion pictures and other audiovisual works; and
(7) sound recordings.

It has been held that the category of § 102(a) into which computer programs fit is the “literary works” category. Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1249 (3d Cir. 1983), cert. dismissed, 464 U.S. 1033 (1984). Thus Intel argued that the Intel Microcode was copyrightable as a literary work. Of course, once it is shown that the Intel Microcode does not qualify as a “computer program” that ground for finding that it is a “literary work” disappears. Intel offered no evidence at trial that its Microcode constituted any other form of literary work, and never seriously advanced any such argument. In any
tic approach assumes the very conclusion being sought and ignores the many significant differences between the Intel Microcode and computer programs.

Rather than simply asking whether the label "computer program" can be made to fit the Intel Microcode, a proper resolution of this crucial question requires an analysis of the function of the Intel Microcode, its relation to the 8086/8088 microprocessors, and the respective concerns and policies sought to be protected and advanced by the patent and copyright laws. Only through such an analysis can one intelligently decide the proper form of protection for the Intel Microcode.

Patents protect new and useful processes (including methods), machines, manufactures and compositions of matter and new and useful improvements thereof.\textsuperscript{29} In order to obtain a patent, an inventor must disclose the invention to the United States Patent and Trademark Office\textsuperscript{30} and meet certain criteria, for example, that the invention be novel, useful and nonobvious.\textsuperscript{31} After a rigorous examination,\textsuperscript{32} to ensure the invention meets the criteria established by Congress, a patent is granted for a limited period of time, namely 17 years.\textsuperscript{33} After expiration of the 17-year period, the invention falls into the public domain. However, during the life of the patent, the inventor has the legal right to exclude others from making, using, or selling the invention.\textsuperscript{34} A patent essentially grants to the inventor a monopoly over the invention expressed in the claims of the patent, without regard to whether a later user copied from the patent or independently developed the same invention. Moreover, under the "doctrine of equivalents", a patent is infringed by any product or process which performs substantially the same function in substantially the same way to obtain the same result, even though the infringing product or process may not correspond to the literal terms of the claims of the patent.\textsuperscript{35}

The copyright laws, on the other hand, protect the original

---

34. \textit{Id.}
writings of authors, rather than inventions. 36 The copyright laws do not protect any idea, but only the author's particular expression of the idea. 37 Moreover, even if a second author produces a work whose expression is identical to the work of another, but is the result of the second author's independent efforts, there is no infringement. 38 To obtain a copyright, nothing need be done except to fix the original work of authorship in a tangible medium of expression. 39 That is, merely by writing the creative expression on a piece of paper, one has a copyright. In order to preserve the copyright, the author must give notice to the world of the copyright by putting a copyright notice on published copies of the work. 40 To enforce a copyright, the owner generally need only obtain a certificate of registration from the Copyright Office. 41 None of the examination procedures utilized for patents applies to copyrights and a copyright lasts for at least three times the 17-year duration of a patent. 42

Both patents and copyrights exist primarily for the public interest, not the interests of inventors and authors. 43 A limited monopoly is granted primarily not as a reward, but to encourage new works for the ultimate public benefit. 44 Given their common origin 45 and purpose, the differences between the patent and copyright laws are striking. Those differences reflect long standing policy judgments about the nature of the works involved, the type of activities to be encouraged, and the cost of the monopoly being granted. Broadly speaking, the goal of patent law is technological progress. Therefore, only those inventions that advance the state of the art in some useful and nonobvious way are protected. On the other hand,

42. See generally, 17 U.S.C.A. § 302 (1977). With certain exceptions, copyright endures for a term consisting of the life of the author plus fifty years. "Works made for hire" (i.e., works created by employees) and certain other works have a duration of at least 75 years.
44. Id.
45. Both the copyright and the patent laws find their origins in Article I, Section 8 of the Federal Constitution, which grants Congress the power: To promote the Progress of Science and useful Arts, by securing for limited times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries.
copyright law encourages a diversity of original, but not necessarily better, writings.

Science and technology are centripetal, conducing toward a single optimal result. One water pump can be better than another water pump, and the rule of patent and trade secret law is to direct investment toward such improvements. Literature and the arts are centrifugal, aiming at a wide variety of audiences with different tastes... The aim of copyright is to direct investment toward abundant rather than efficient expression.46

To relate the foregoing concepts to the Intel Microcode, some understanding of computer technology is required. Computers in general, including microprocessors, consist basically of five parts; an arithmetic logic unit, registers, a control unit, input and output, and main memory. A computer is a machine that is designed to perform certain specified operations, i.e., those operations specified in its instruction set.47 In early computers, each operation was performed entirely by hardware. In the early 1950s, however, English designers concluded that computer design could be simplified through the use of microcode as the control unit of the computer.

To employ microcode in a computer, the computer designer defines a series of suboperations, and organizes the control unit of the computer so that each macroinstruction is performed by activating the appropriate sequence of microoperations and appropriate logic circuits. Thus, microcode can be seen as a sequence of instructions, but at a level indistinguishable from the hardware. The major advantage of microcode is simplicity of design. A microcoded microprocessor can be designed in less time than a microprocessor without microcode. The price paid for such an advantage is some loss of performance, i.e., speed of execution.

The Intel Microcode is a fixed part of the control portion of the 8086/8088 microcomputer, and indeed, in all meaningful respects is inseparable from it. The 8086/8088 does not exist without that Intel Microcode portion any more than it would exist if its arithmetic-logic unit (ALU) section were to be removed. Correspondingly, the Intel Microcode has no practical use other than as a functioning part of the entire 8086/8088 microprocessor. Because of this close tie between microcode and hardware, the form of the microcode is dictated more by the requirements of the computer than by the creativity of an author. Thus, while much of the same terminology

47. See supra note 7.
which describes computer programs is, unfortunately, also used to describe microcode, the function that microcode performs is the function performed by hardware and indeed the Intel Microcode takes the form of hardware (i.e., transistors) in the Intel 8086/8088.

Accordingly, because microcode is effectively inseparable from the microprocessor itself, the sensible approach in determining the legal protection to be afforded microcode is to decide on the proper form of protection for the microprocessor as a whole, including the microcode. Microprocessors are already subject to patent\textsuperscript{48} and mask work\textsuperscript{49} protection. To now extend, in addition, copyright protection to such a device, even if ostensibly limited to its microcode portion, may well produce de facto copyright protection for the entire microprocessor, a result clearly not envisioned by the copyright laws.\textsuperscript{50} Therefore, copyright protection for microcode cannot be discussed without considering the extent to which its companion hardware will thereby also be "protected". If the professed goal in seeking copyright protection for microcode is to encourage new and better microprocessors, then patent and mask work law already provide appropriate and sufficient protection.

At trial, Intel never advanced any substantive policy reasons to justify microcode copyrightability. Rather, Intel's strategy was to mechanically show that its Microcode bore similarities to computer software and argue that it therefore fit the statutory definition of 17 U.S.C. Section 101: A "computer program" is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result. Intel argued that its Microcode is a computer program because, like a computer program, it can be represented by zeros and ones, can be stored on a computer tape, and can be expressed in high and low level symbolic languages. However, these analogies miss the point. The entire microprocessor, or indeed any hardware, can be represented in zeros and ones, and that representation can be stored on a computer tape, and can be expressed in high and low level symbolic languages. Every machine and process, no matter how complex, is subject to such representation.\textsuperscript{51} It is clear, however, that copyright law protects only such representation, not the thing represented.\textsuperscript{52}

\textsuperscript{50} See supra note 22. By the end of the NEC v. Intel trial, this is precisely what Intel had tried to do. See infra notes 171 to 179 and accompanying text.
\textsuperscript{51} Tr. XII 1861:7-1862:1.
\textsuperscript{52} Imperial Homes Corp. v. Lamont, 458 F.2d 895 (5th Cir. 1972) (copyrighted architectural plans cannot clothe author with exclusive right to reproduce the dwelling pictured);
In response to Intel’s semantic argument, NEC pointed out that the Copyright Act’s definition of a computer program does not, as urged by Intel, require a finding that the Intel Microcode is a computer program. On the contrary, that definition limits computer programs to sets of statements of instructions “to be used . . . in a computer.” The definition itself plainly shows Congress’ underlying premise that, before there can be a computer program, there must first be a computer in which the program can be used; the computer and the computer program are two distinct and separate entities. When so understood, the Intel Microcode does not meet the definition of computer program because it is part of the computer itself, rather than something used “in a computer.” Without the microcode, there is no computer.

The question of whether microcode is “used in a computer” or is “part of the computer” is more than mere semantics. If microcode is an integral part of a noncopyrightable computer and has no use other than as a part of such computer, what function would copyright protection serve? Unlike software, microcode is highly hardware dependent. Microcode cannot be moved from one machine to another. The Intel Microcode, for example, could never be made to work on the NEC microprocessor, even though the microarchitectures have similarities.53

At trial, Intel relied heavily on the Apple v. Franklin54 case, which held that an operating system program was copyrightable. However, the function of the Intel Microcode is quite different from the role played in a computer by either an application program or an operating system program.55 Even the latter, although necessary to allow a less sophisticated user to operate the computer, is nonetheless not a part of a computer. A computer can exist without any specific operating system and the same operating system can be used in different computers.56 This is not the case with the Intel

Russell v. Trimfit, Inc., 428 F. Supp. 91 (E.D. Pa. 1977), aff’d., 568 F.2d 770 (3d Cir. 1978) (even if plaintiff’s concept of mitten toe socks or glove socks were original creation, copyrights and copyrighted drawings of socks did not protect against defendant’s manufacture of such socks).

53. Even the microcode in the V20/V30 had to be modified to work in the V40/V50, although the microarchitectures were nearly identical. Compare NEC Ex. 179 and 179A.
  55. “Application programs” perform a specific task for a computer user, such as word processing, checkbook balancing, or playing a game. “Operating system programs” manage the internal functions of the computer or facilitate use of application programs.
  56. A computer and its operating system constitute two products. In re Data General Corporation Antitrust Litigation, 490 F. Supp. 1089 (N.D. Cal. 1980), aff’d in relevant part, 734 F.2d 1336 (9th Cir. 1984), cert. denied, 105 S. Ct. 1334 (1985). It is clear on the other
Microcode portion of the 8086/8088 microprocessors.57

Ultimately, the question of copyrightability must focus on the monopoly being sought, and the cost of such a monopoly. Congress and the courts have viewed with suspicion efforts to extend copyright to protect industrial products. This is shown, for example, in the Copyright Act's provisions relating to a useful article and its design or shape. The term "useful article" is defined in 17 U.S.C. Section 101 as "... an article having an intrinsic utilitarian function that is not merely to portray the appearance of the article or to convey information." Although sculptural works are copyrightable58 and the shape of any article, including a useful one, may be characterized as a sculptural work, Congress chose to regulate the interplay between copyright protection and utilitarian devices by limiting copyright protection for the design of useful articles.

Such [pictorial, graphic and sculptural] works shall include works of artistic craftsman insofar as their form but not their mechanical or utilitarian aspects are concerned; the design of a useful article, as defined in this section, shall be considered a pictorial, graphic or sculptural work only if, and only to the extent that, such design incorporates pictorial, graphic or sculptural features that can be identified separately from, and are capable of existing independently of, the utilitarian aspects of the article.59

This denial of copyright protection to industrial designs prevents de facto monopolies over noncopyrightable utilitarian articles. A design which cannot be identified separately from, and is incapable of existing independently of, the utilitarian aspects of the article must be available for competitors to use, or they will be unable to produce the underlying utilitarian article.60 The same consideration obtains in the case of the Intel Microcode. The extension of copyright protection to that portion of the 8086/8088 microprocessor may effectively prevent competitors from using even the hardware of Intel's microprocessors, because the microcode and the hardware are so intimately related and mutually dependent.61

57. Indeed, Apple v. Franklin specifically distinguished microcode from computer programs and stated that microcode was not involved in its decision. 714 F.2d 1240, 1249 n. 7 (3d Cir. 1983).
60. See Esquire, Inc. v. Ringer, 591 F.2d 796, 800-801 (D.C. Cir. 1978).
61. Indeed, Intel's ultimate position at trial was that NEC should not have used hardware similar to that which had been used by Intel. See infra notes 171 to 179 and accompanying text.
The cost to society of a copyright monopoly for industrial products may be high. In some circumstances copyright may well discourage the creation of new and better products. In many industries companies study competitors' products, learn from them, and use that knowledge to develop better products. Under traditional copyright law, suppressing a competing industrial product would be impossible. Section 102(b) of the Copyright Act makes clear that copyright protection extends only to an author's expression:

In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work.

Therefore, for example, one cannot prevent a competitor from making and selling a similar machine by copyrighting the machine's operating manual which describes how the machine works.

Yet that is precisely Intel's purpose. There was nothing particularly unusual or unique about the Intel Microcode. What makes the Intel Microcode important is that its design is inextricably tied to the hardware of the 8086. How closely tied the two were became apparent when Intel raised its "bootstrap license" argument, asserting in effect that its alleged microcode copyright could also protect the 8086 architecture. The copyright monopoly Intel seeks would prevent the study of existing technology and thereby retard innovation in the market place, which innovation should in fact be encouraged. It is important to distinguish between two situations: (a) defendant studies the plaintiff's product to learn how it works, and then, using that knowledge, designs his own similar but improved product; (b) defendant studies the plaintiff's product, but to save time and effort copies it with minimal changes. Whatever one feels about the second case, it would be a serious error to create a system for protecting legal rights that discourages the first. Yet the system urged by Intel, in which the paramount issue is whether the defendant had knowledge of the plaintiff's product certainly penal-

62. See supra note 12.
65. Tr. IV 557:3-12; 565:23-566:25.
66. See infra notes 171 to 179 and accompanying text.
67. See supra note 12.
izes severely any study of competing products.\(^6\)

If the Intel Microcode is found to be copyrightable, the ultimate effect may be to defeat the strong public interest in free access to technology and to stifle legitimate competition. Any competitor wishing to offer a functionally compatible computer or microprocessor could easily be charged with copyright infringement by the original developer solely on the basis of similarity in function. The copyright monopoly, with its presumption of validity,\(^6\) would be a frightening weapon having a significant \textit{in terrorem} effect against any competitor developing a compatible computer, whether copied or not.

To summarize, courts should not be misled by a simplistic characterization of microcode as a computer program into concluding that microcode is copyrightable. To do so is to give labels and semantics supremacy over substance. There is much to lose and nothing to gain by extending copyright protection to structures that are inseparable from hardware.

\section*{B. Forfeiture Due to Lack of Notice}

A second major issue at trial was whether Intel had forfeited its alleged copyrights through the failure of Intel's licensees to affix copyright notice to copies of the Intel Microcode which they distributed. To maintain copyright protection in the United States, every copyright owner is required to give public notice of the claim of copyright in the form and manner prescribed by statute on copies distributed to the public under its authority.\(^7\) 17 U.S.C. Section 401(a) reads in pertinent part:

\begin{quote}
Whenever a work protected under this title is published in the United States or elsewhere by authority of the copyright owner, a notice of copyright as provided by this section shall be placed on all publicly distributed copies from which the work can be visually perceived, either directly or with the aid of a machine or device.
\end{quote}

Failure to affix the required copyright notice will result in loss of copyright protection unless the omission is otherwise permitted, excused or timely corrected pursuant to some other section of the

\(^6\) The basic principle is stated in the CONTU Report as follows: programmers are free to read copyrighted programs and use the ideas embodied in them in preparing their own works. Final Report, National Commission on New Technological Uses of Copyrighted Works (1978) at 51. \textit{See infra} note 169.

\(^7\) 17 U.S.C.A. § 410(c) (1977).
The statutory saving provisions that were the primary focus of evidence and argument in NEC v. Intel were two provisions of 17 U.S.C. Section 405(a). As relates to the evidence in the case, those subsections provide that copyright protection will not be lost, notwithstanding the distribution of copies without proper notice affixed, if either (i) the number so distributed was “relatively small”; or (ii) a reasonable effort is made by the copyright owner to have proper notice added to copies distributed within the United States after discovery of the omission.

NEC showed that there had been distribution of copies of the “work” (i.e., 8086/8088 microprocessors containing the Intel Microcode) under Intel’s authority without notice. Intel admitted and the evidence at trial showed that millions of microprocessors containing the Intel Microcode had been distributed by at least four Intel licensees, including NEC, to the public without any copyright notice whatsoever. The licenses from Intel to those four licensees contained no requirement that the licensees affix copyright notice to the microprocessors containing the Intel Microcode which they distributed.

This showing by NEC established a prima facie case of Intel’s failure to comply with the statutory notice requirement. It was Intel’s burden to prove a defense under Section 405(a)(1) or (2); that the quantities distributed were relatively small, or that Intel had made a reasonable effort to add notice to copies distributed in the

---


72. 17 U.S.C.A. § 405(a) (1977) reads as follows:

   Effect of Omission on Copyright. The omission of the copyright notice prescribed by sections 401 through 403 from copies or phonorecords publicly distributed by authority of the copyright owner does not invalidate the copyright in a work if

   (1) the notice has been omitted from no more than a relatively small number of copies or phonorecords distributed to the public; or

   (2) registration for the work has been made before or is made within five years after the publication without notice, and a reasonable effort is made to add notice to all copies or phonorecords that are distributed to the public in the United States after the omission has been discovered; or

   (3) the notice has been omitted in violation of an express requirement in writing that, as a condition of the copyright owner’s authorization of the public distribution of copies or phonorecords, they bear the prescribed notice.

73. Intel Exs. A, EV, EX; NEC Ex. 254. Accordingly, the safe harbor of § 405(a)(3) was not available to Intel.

74. Cooling Systems and Flexibles, Inc. v. Stuart Radiator, Inc., 777 F.2d 485, 489-490 (9th Cir. 1985); Canfield v. Ponchatoula Times, 759 F.2d 493, 499 (5th Cir. 1985).
United States\textsuperscript{75} after discovery of the omission.

1. The Number Without Notice Was Not "Relatively Small"

17 U.S.C. Section 405(a)(1) provides that distribution without copyright notice does not invalidate one's copyright if "the notice has been omitted from no more than a relatively small number of copies . . . distributed to the public." NEC introduced evidence showing that approximately 11,000,000 8086/8088 microprocessors embodying the Intel Microcode were distributed without proper notice by Intel licensees, which represented about half of the total number of copies distributed. Intel disputed this number but admitted that about 3,000,000 microprocessors (or 11 percent of the total as calculated by Intel) had been distributed without notice under its authority, and urged that 3,000,000 was a relatively small number.

Most of the disparity in the numbers resulted from the different manner in which the parties classified microprocessors which had been distributed bearing forms of copyright notice other than "© Intel 1978." For example, it appeared that one of Intel's licensees, Advanced Micro Devices (AMD), distributed over five million devices bearing the notice "© AMD 1982." An interesting dispute therefore arose regarding whether these copies should be counted as marked or unmarked.\textsuperscript{76}

\textsuperscript{75} For the purpose of determining whether the notice requirement was violated, distribution throughout the world is considered (17 U.S.C.A. § 401(a) (1977)), but to determine whether there had been a reasonable effort to correct the omission after discovery, the test is what was done for copies distributed in the United States.

\textsuperscript{76} Intel's certificates of copyright registration showed that the year of first publication of the Intel Microcode was 1978. The proper form of copyright notice was therefore "© Intel 1978." Although § 406(a) provides that an error in the name of the copyright owner appearing in the notice does not affect the validity of the copyright, 17 U.S.C.A. § 406(b) (1977) requires that:

Where the year-date is more than one year later than the year in which publication first occurred, the work is considered to have been published without any notice and is governed by the provisions of Section 405. [Emphasis added].

Therefore, NEC argued, the AMD devices, among others, should be classified as unmarked for purposes of determining the percentage of unmarked devices, because 1982 (the year-date used by AMD) was more than one year later than 1978, the year in which publication of the Intel work first occurred.

In response, Intel argued without any support in the record, that the form of notice employed by AMD was in fact proper because it referred to AMD's "copyright" in the topography or layout of the chip, which was first "published" in 1982. Intel argued further, that because the AMD topography was a "derivative work of the Intel Microcode", the single AMD notice with the later year-date was effective under § 401(b)(2) to protect both the topography and the earlier-published Intel Microcode. 17 U.S.C.A. § 401(b)(2) (1977) provides that:

[I]n the case of compilations or derivative works incorporating previously pub-
Intel's Section 405(a)(1) defense to copyright forfeiture raised several other interesting questions for which no definitive case law answer could be found. For example, does the phrase "relatively small number" as used in 17 U.S.C. Section 405(a)(1) mandate consideration solely of the percentage of devices distributed without notice, as urged by Intel, or should consideration also be given to the absolute number distributed? At what point in time should the published material, the year date of first publication of the compilation or derivative work is sufficient.

In reply, NEC first pointed out that there was no evidence in the record to support Intel's derivative work theory. Furthermore, NEC argued that apart from Intel's failure of proof, Intel's theory was patently wrong because (i) topography is in no meaningful sense "derived" from microcode, and (ii) even prior to the passage in 1984 of the Semiconductor Chip Protection Act, 17 U.S.C.A. §§ 901-914 (West Supp. 1986), which provided an exclusive and sui generis form of protection to a chip's layout for a ten-year period, topography had never been held to constitute copyrightable subject matter and all attempts to register such "works" had been uniformly rejected by the Copyright Office.

Judge Ingram ultimately held, without explanation, that "[c]ontentions of mismarking... and purported issues relating thereto are not supported in the evidence." 645 F. Supp. 590 (N.D. Cal. 1986), Partial Finding of Fact No. 53. This finding suggests a view by Judge Ingram that NEC's actual burden was not only to show that the notice on the AMD chips deviated more than one year from the date of first publication of the Intel Microcode, but also to anticipate and affirmatively disprove any possible theory which would validate the AMD notice under the copyright law.

Subsequent to the announcement of the Partial Findings, the Ninth Circuit adopted NEC's position that the burden of demonstrating that the AMD notice was valid had been on Intel. Lifshitz v. Walter Drake & Sons, Inc., 806 F.2d 1426, 1432 (9th Cir. 1986). In effect, however, the entire issue of the mismarked chips became moot because Judge Ingram also held that even the number of copies admitted by Intel to have been distributed without notice was not "relatively small," and thus § 405(a)(1) failed to provide a defense to copyright forfeiture.

77 In the cases to date, the largest absolute number which has been held to have been "relatively small" was approximately 400, which amounted to 1% of the total number distributed. Original Appalachian Artworks, Inc. v. Toy Loft, Inc., 684 F.2d 821, 827 (11th Cir. 1982). In a subsequent case, the Eleventh Circuit commented on its opinion in Original Appalachian Artworks as follows:

We disagree, however, with the assertion that the holding in Original Appalachian Artworks to the effect that 1 percent (400) of the total number of copies meet the "relatively few" test, necessarily contradicts a conclusion that 1,000 copies amounting to 1 percent of the total number in another situation would not meet the test. The question must be answered on a case-by-case basis in light of the totality of the circumstances. We find that on the facts of this case, the 2,500 copies of the Hollybrooke in the promotional brochure and advertising folders, which totaled approximately 2.4 percent of the total number of copies, constitutes more than a relatively small number. Unlike the situation in Original Appalachian Artworks where the 1 percent had defective notices, the copies we are considering in the instant case had absolutely no indication of copyright affixed. Furthermore, "2,500 copies is a significant number in the absolute sense." Hence we find that the savings provision of 405(a)(1) does not apply to the Hollybrooke either. [Emphasis added.]

Donald Frederick Evans and Associates, Inc. v. Continental Homes, Inc., 785 F.2d 897, 910 (11th Cir. 1986). See also, Florists' Transworld Delivery Association v. Reliable Glassware
comparison be made between copies with notice and copies without notice? If the number distributed without notice is, at any point in time, more than relatively small, is the excuse under Section 405(a)(1) forever lost, or may a number which at Time X is relatively large (for instance 50%), later become at Time Y relatively small, (for instance 1%) by flooding the market with properly marked copies? Should copies distributed by licensees who were under an obligation to affix notice but whose products in fact contained no notice be counted as bearing proper notice, no notice, or were they to be totally excluded from consideration?

It is unclear from the Partial Findings whether Judge Ingram considered these questions and if so, how they were decided. Intel admitted that more than three million devices (constituting approximately 11% by calculations most favorable to Intel) were distributed without notice. The Court found this undisputed number to be more than relatively small and Intel's Section 405(a)(1) defense was rejected.

2. Efforts to Add Notice After Discovery Were Not Reasonable

Alternatively, Section 405(a)(2) provides that publication without notice will not invalidate the copyright if:

registration for the work has been made before or is made within five years after the publication without notice, and a reasonable effort is made to add notice to all copies or phonorecords that are distributed to the public in the United States after the omission has been discovered.

Intel urged that, after discovering the omission of notice, it had taken reasonable steps to have notice added to copies thereafter distributed in the United States and therefore that the omission of copyright notice from the prior distributions should be excused under Section 405(a)(2). The timeliness of Intel's copyright registration was not an issue. However, in order to determine the reasonableness of Intel's efforts it was first necessary to determine when discovery took place so that Intel's efforts could be evaluated from that date.\(^{78}\)

and Pottery Co., Inc., 213 U.S.P.Q.(BNA) 808 (N.D. Ill. 1981) (§ 405(a)(1) inapplicable "[b]ecause almost one million vases were sold to member florists without the affixed notice," without any discussion by the court of percentages.)

\(^{78}\) The test set forth in 17 U.S.C. § 405(a)(2) of "reasonable effort" has been, without exception, held to require not only reasonable acts but also reasonable acts within a reasonable time. Gemveto Jewelry Company Inc. v. Jeff Cooper, Incorporated, 568 F. Supp. 319 (S.D. N.Y. 1983) (nothing done for "a minimum [of] several months," and possibly longer,
a. **Discovery**

The evidence presented by Intel with respect to discovery\(^79\) was primarily the testimony of Intel Vice President David House. Mr. House testified that discovery occurred some time in the Spring of 1985, as then reported to him by the Intel Legal Department.\(^80\) He said he was assured that efforts were being made to have notice added and that he believed that letters were written to licensees and stickers containing proper notice were sent out to distributors at about that time.\(^81\)

Evidence introduced by NEC showed that Intel's in-house General Counsel had asserted in writing that Intel knew of NEC's nonmarking as early as August 1984.\(^82\) Further, Intel engineers routinely examined microprocessor products of competitors, including the nonmarking licensees, and such examination had to have shown as early as 1981 that those products bore no copyright notice either on the package or on the die.\(^83\) NEC also introduced testimony that reports written by those Intel engineers regarding their

---

\(^79\) The term "discovered" is not defined in the Copyright Act or in the cases that have construed 17 U.S.C. § 405(a)(2). "Discovery", however, is generally held to include notice or information of circumstances which would put a reasonable person on notice. In Donald Frederick Evans and Associates, Inc. v. Continental Homes, Inc., 785 F.2d 897 (11th Cir. 1986) nothing done for ten months not reasonable)

\(^80\) Tr. V 694:7-22; 695:11-25.

\(^81\) Tr. V 706:9-20; 719:4-12; 748:14-749:8.

\(^82\) NEC Ex. 367.

\(^83\) Tr. 2303:7-16; NEC Ex. 33. The die is the actual integrated circuit, a small piece of silicon on which the microprocessor is etched. The package is a sealed container in which the die is mounted.
examinations of competitive products were distributed to Intel's senior management. In addition, there was testimony that Intel knew or should have known at the time that at least one of the 8086/8088 licenses were executed, that the particular licensee was not then affixing notice of Intel's copyright.

There is no case law that considers the relationship between Section 405(a)(3) and Section 405(a)(2) of the Copyright Act. NEC argued that because Intel had failed to expressly require licensees to mark, it had some duty to monitor those licensees to see whether they were marking the 8086 products manufactured under license from Intel. No mention of this argument is made in Judge Ingram's Partial Findings.

In addition, Judge Ingram apparently did not consider that what Intel's engineers must have known was thereby known to Intel itself, for Judge Ingram agreed with Mr. House that Intel did not discover the lack of notice until the spring of 1985 when its Legal Department commissioned an engineer to purchase NEC and other devices to examine them for notice.

b. Efforts

The second issue faced by the Court in determining whether the omission of copyright notice was to be excused under Section 405(a)(2) was whether Intel's actions upon discovery of the absence of notice constituted the "reasonable effort" contemplated by that subsection. Although the courts have tended to find a reasonable effort wherever possible, even those courts which have been most protective of copyright owners have required that at least some action be taken after discovery to prevent future distribution without notice in the United States.

The "effort" taken by Intel to have notice added to copies subsequently distributed in the United States consisted essentially of two steps. First, in the spring of 1985, long after Intel's employees had examined the licensees' products and after NEC had already asserted forfeiture of the Intel copyrights due to omission of notice, Intel wrote letters to the licensees which had not been marking. Intel's second step was to send small red stickers saying © Intel

84. NEC Ex. 33.
85. Tr. IV 603:24-605:12.
86. See supra note 72.
88. See 17 U.S.C.A. § 405(a) (1977) which is reproduced in note 72, supra.
89. See, e.g., Shapiro & Son Bedspreads Corp. v. Royal Mills Assoc. 764 F.2d 69 (2d Cir. 1985); Canfield v. Ponchatoula Times, 759 F.2d 493, 499 (5th Cir. 1985).
1978” to NEC’s distributors demanding that the stickers be placed on all unmarked devices without offering to recompense the distributors for the expense involved or for the risk of damage to those devices during such stickering.

The letters sent by Intel to its licensees were deliberately misleading. Although Intel knew that certain licensees were not marking and had no obligation to mark under the terms of their respective licenses, Intel nonetheless asked them to “confirm” that they were complying with “their obligation” to mark.90 The letters did not ask the licensees to start marking, nor offer to reimburse them for the cost of marking which the licensees were under no obligation to do.

The misleading nature of the letters prompted concern and consternation on the part of the licensees and actually delayed initiation of marking. The licensees, quite naturally, pointed out that there was no marking obligation in their license agreements. After several months of communications, two of the licensees eventually agreed to add notice to new product beginning some time in the middle of July. Products already in distribution in the United States but which had not yet reached the public would only be marked as and when convenient. At no time did Intel ever offer to reimburse any licensee for expenses incurred.

The only other action taken by Intel and urged by it as being reasonable for purposes of 17 U.S.C. Section 405(a)(2) was the mailing of hundreds of thousands of small red stickers saying “© Intel 1978” to NEC’s distributors91 together with the request that the distributors place a sticker on each microprocessor. One major distributor testified that it simply ignored the request since the cost would be considerable and there would be a substantial risk of damaging individual devices in the process of putting stickers on them.92 When that distributor called the Intel Legal Department to com-

90. The letters also stated that “[o]n a regular basis, we ask our licensees to certify that the appropriate markings are in fact being made.” The head of Intel’s Business Development Department testified, however, that there was no on-going audit program; these were the first such letters. Tr. XVII 2570:23-2571:9. This illustrates further that Intel was only interested in creating a paper trail for purposes of the present lawsuit.

91. Intel did not send any stickers to other licensees to affix to unmarked products until October, 1985, at the earliest. Intel never sent stickers to distributors of those licensees; those distributors continued selling unmarked product for many months, until their inventories were exhausted. Tr. VI 910:25-912:19.

92. Tr. XIX 2642:19-2645:19. The human handling of semiconductor devices can easily render them unusable due to bent or broken leads or the transmission of static electricity to the device. Therefore, the distributor testified, employees are discouraged from handling parts.
plain about the effort being requested by Intel, the advice from Intel's in-house counsel was to just "forget it."

In all, Intel's actions were designed to create an inexpensive paper trail while shifting the cost of marking to licensees or distributors who were under no obligation to mark. That Intel's efforts were ineffective is shown by the fact that there was a delay of several months following the letters before Intel's licensees started to mark new product, and apparently not one sticker was ever placed on an existing product by a distributor or licensee. It is also a matter of record that Intel never checked to see whether any of the stickers which had been sent out had in fact been used. Intel knew that the stickers would not be used; the effort was a sham. The attitude of Intel toward the stickers was confirmed by the testimony of one of Intel's in-house attorneys that when he heard of the stickers, he congratulated Intel's General Counsel for a great tactical stratagem.

Effective means of securing marking were open to Intel. The minimal reasonable efforts required of Intel were: (i) to have offered to bear the expense of immediate remarking of all microprocessors under control of its licensees (Intel did not do this); (ii) to have offered to bear the expense of replacing unmarked microprocessors still in the inventories of the licensees' distributors with units which were properly marked (Intel did not do this); and (iii) to have offered to bear the expense of its licensees redoing their masks (used in the manufacturing process) so that future microprocessor production would have the proper copyright notice on the semiconductor itself (Intel did not do this). These minimal reasonable efforts were required even though some extra effort and expense might have been incurred.

Intel's failure to comply with the provisions of 17 U.S.C. Section 401 was not the error of an innocent and unsophisticated copy...

93. Tr. XIX 2646:3-18.
94. Tr. V 753:18-20; VI 909:19-25.
95. Tr. VI 909:19-25.
96. Tr. XVIII 2578:10-2582:9. Of course, in other situations, sending out stickers or tags to be added by distributors has been held to be part of a reasonable effort. See, e.g., Shapiro and Sons Bedspreads v. Royal Mills Assoc., 764 F.2d 69 (2d Cir. 1985); Florists' Transworld Delivery Ass'n v. Reliable Glassware and Pottery, Inc., 213 U.S.P.Q.(BNA) 808 (N.D. Ill. 1981). However, in those cases the products involved were relatively few and not susceptible to damage by being handled.
97. "Implicit in the concept of a 'reasonable effort' under § 405(a)(2) is the expectation that an expenditure of time and money over and above that required in the normal course of business will be made." Videotronics, Inc. v. Bend Electronics, 586 F. Supp. 478, 483 (D. Nev. 1984).
right owner. Intel is knowledgeable with respect to patents and copyrights and aggressively uses intellectual property laws in support of its marketing efforts. During the time Intel was ignoring its obligations under 17 U.S.C. Section 401, Intel was fully aware of this requirement and contemporaneous internal Intel memoranda comment on the significance and consequences of failure to use proper copyright notice. Notwithstanding this, Intel failed to take the simple precautionary steps of inserting written obligations to mark in its license agreements, which would have given Intel an absolute safe harbor under 17 U.S.C. Section 405(a)(3). Yet, Intel has so far been able to avoid the consequences of its conduct with an ineffective series of letters and stickers designed to pass on to others the cost of the reasonable effort to add notice.

C. Noninfringement

The third major issue at trial was whether, if Intel had valid copyrights on its Microcode, the NEC Microcode infringed those copyrights.

1. The NEC and Intel Microcodes

Microcode, as used in the microprocessors at issue, can be described as the control structure for implementing a given set of macroinstructions on a given hardware. Microcode can be represented in several different forms. At the machine level, it is expressed by the presence or absence of transistors formed as part of the microprocessor; in object code form it can be represented by zeros and ones; it can also be represented in source code by a set of microinstructions. When two computing machines execute the same macroinstruction set on the same or similar hardware, there will be certain similarities in the functioning of the two microcodes. It is these functional similarities between the NEC and Intel Microcodes that exist as a result of the constraints imposed by the macroinstructions and hardware that Intel focused on during the trial.

At the outset, it should be noted that physically the Intel and NEC Microcodes are quite different in almost every meaningful way. The Intel Microcode is 512 lines long and each line is 21 bits wide. The NEC Microcode is 1024 lines long, and each line

98. NEC Ex. 246, p. 164.
99. See supra text accompanying note 47.
100. See infra notes 121 to 164 and accompanying text.
101. A "bit", commonly represented by a zero or a one, is the smallest unit by which
is 29 bits wide. The difference in line width results from several basic differences in format between the two Microcodes. These format differences arise both from NEC's unique dual bus architecture and from the existence of unique fields in the NEC Microcode. In addition, the Intel Microcode has six different types of microinstructions available whereas the NEC Microcode has only three types available. The object codes (the pattern of zeros and ones) and the source codes (the microinstructions in human readable form) which describe the physical Microcodes are also totally different.

Not all of the 1024 lines in the NEC Microcode are used for the 8086 instruction set. Of the total 1024 lines, 472 were devoted to other purposes: 168 to run original NEC instructions which are not found in any Intel products; 116 for the Intel 80188/80186 instructions which can be run on the V-Series; and 188 for emulation of the 8080 microprocessor. Intel made no claim that any of these parts of the NEC Microcode infringed any Intel copyright. The remaining 552 lines of the NEC Microcode are devoted to carrying out the 8086 instruction set and are the focus of Intel's allegations.

Given the near total dissimilarity between the Intel and NEC Microcodes, Intel resorted to a "translation" of the two Microcodes to a hybrid "common language" in order to show "similarities" at trial. While a translation may be useful to facilitate a comparison, Intel's translation, done in a language not used by either of the original Microcodes, eliminated or actually hid many of the real differences between the two Microcodes which result from hardware or other differences.

For example, where the Intel Microcode required two operations to move information from two memory locations to an adder,
NEC could move such information in parallel because of its unique dual bus structure. Through its translation, Intel eliminated differences arising from the NEC structure. Of course, by imaginative creation and application of translation rules such as those proposed by Intel it is possible to eliminate any difference whatsoever given sufficient ingenuity on the part of the person doing the translation. By means of such translation rules Intel was able to create similarities where none had existed previously.

Intel's translation was questionable in other respects as well. For example, Intel did not include in its translation those Intel sequences that did not have a corresponding NEC routine. Most notable was the "magic instruction", an undocumented instruction hidden by Intel in the Intel Microcode to detect copying.\(^{107}\) When Intel analyzed the NEC Microcode it found, to its disappointment, that it did not contain the magic instruction.\(^{108}\) Because the NEC Microcode did not contain the magic instruction, Intel did not translate it. Similarly, Intel used microcode to perform effective address generation, while NEC elected instead to use hardware to perform that operation. Intel did not include the effective address sequence in its translation of the Intel Microcode.

The creation of similarities between the Microcodes through Intel's translation was compounded by the approach taken at trial by Intel's expert witness, Dr. David Patterson. In his comparison of the Microcodes, Dr. Patterson ignored differences between the Microcodes if the differences were required by differences in hardware or by differences in the NEC and Intel microinstruction sets, that is, where both the 8086 and V-Series microcode designers were not able to make the same choice.\(^{109}\) Accordingly, Intel took the contradictory position of complaining that NEC had used similar hardware to facilitate copying,\(^{110}\) but where there were differences in NEC's hardware or microinstruction set, Intel ignored the resulting differences in the NEC Microcode.

Obviously, NEC disputes the validity of Intel's translation rules, as well as Dr. Patterson's methodology and analysis. The re-
remainder of this article, however, shows that even after the two Microcodes were translated by Intel, they still were not substantially similar in protected expression.

2. The Law of Copyright Infringement

Copyright infringement consists of two elements: (a) proof that the defendant copied from the plaintiff’s copyrighted work and (b) proof that the copying “went so far as to constitute improper appropriation.” It is not enough to prove merely the first element, i.e., that the defendant copied, because some types of copying are permissible. The second element of every copyright infringement case, showing what and how much was copied, is often referred to as the substantial similarity test whereby the plaintiff must show that the expression utilized by the defendant is substantially similar to the protected expression in plaintiff’s work. As described infra, Intel tried to structure its case to avoid the second part of the Arnstein test. Intel’s approach was to point to isolated similarities between the two Microcodes in an attempt to convince the Court that NEC had copied something from Intel, and then try to equate such copying with infringement. As a consequence, it was, conveniently to Intel, never made clear exactly what protected expression Intel claimed NEC had copied.

3. Intel’s Isolated Similarities Case

Regardless of how a plaintiff attempts to prove the first element of the Arnstein test, every plaintiff must also prove “substantial similarity of protected expression” between the accused work and the

111. Arnstein v. Porter, 154 F.2d 464, 468 (2d Cir. 1946); Sid & Marty Krofft Television Productions, Inc. v. McDonald’s Corporation, 562 F.2d 1157, 1164-65 (9th Cir. 1977).
112. Arnstein v. Porter, 154 F.2d 464, 472-473 (2d Cir. 1946); Hoehling v. Universal City Studios, Inc., 618 F.2d 972 (2d Cir. 1980); Landsberg v. Scrabble Crossword Game Players, Inc., 736 F.2d 485 (9th Cir. 1984). Types of permissible copying include copying of noncopyrightable elements of a work such as ideas, merged expression and scenes a faire. See infra notes 165 to 170 and accompanying text. Some courts have carelessly and erroneously assumed that when copying has been proven, there is no need to consider what and how much has been copied. See infra note 122. For an example of a case where a court found “permissible copying”, see Universal Athletic Sales, Inc. v. Salkeld, 511 F.2d 904 (3d Cir. 1975).
113. See, e.g., Sid & Marty Krofft Television v. McDonald’s Corp., 562 F.2d 1157, 1164 (9th Cir. 1977): “If there is substantial similarity in ideas, then the trier of fact must decide whether there is substantial similarity in the expressions of the ideas so as to constitute infringement.”
114. NEC showed that those similarities were not in fact evidence of copying. Rather, they were similarities which had to exist given the constraints on the NEC microcode designer. See infra notes 147 to 164 and accompanying text.
copyrighted work. In *NEC v. Intel*, this required Intel to show the Court how the two Microcodes, which physically appear very different, are in fact substantially similar in protected expression.

Certainly, Intel's translation guaranteed the Microcodes would appear somewhat similar, but that is not sufficient. A showing of substantial similarity requires more than inherent or trivial similarities. It requires that the small part of the Intel Microcode, if any, that constitutes protected expression be distilled and be shown to be substantially similar to the NEC Microcode.

Distinguishing the protected expression, if any, contained within the Intel Microcode from the unprotectible "idea, procedure, process, system, method of operation, concept, principle, or discovery," and determining what part of the expression contained within the Intel Microcode is unprotectible due to the merger doctrine, requires a sufficient technical understanding of microcode.

Rather than educating the Court as to the nature of microcode, Intel's approach was essentially to ignore the required showing of


116. See supra notes 106 to 110 and accompanying text.


[It is not enough to observe that there are a great number of similarities in expression between the two games. It is necessary to determine whether the similar forms of expression are forms of [expression that] simply cannot be avoided in any version of the basic idea of a video game involving space rocks.]

118. Cooling Systems and Flexibles, Inc. v. Stuart Radiator, Inc., 777 F.2d 485, 491 (9th Cir. 1985); Litchfield v. Spielberg, 736 F.2d 1352, 1355 (9th Cir. 1984), cert. denied, 105 S. Ct. 1753 (1985); Sid & Marty Krofft Television Productions, Inc. v. McDonald's Corp., 562 F.2d 1157, 1162 (9th Cir. 1977).

119. The copyright law prohibits only the substantial appropriation of the protected expression of another's work, but does not protect any ideas contained within that work. 17 U.S.C.A. § 102(b) (1977) provides:

In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle or discovery, regardless of the form in which it is described, explained, illustrated or embodied in such work.

120. The "merger doctrine", discussed more fully infra at notes 165 through 170 and accompanying text, states essentially that where there are only one or a few dissimilar ways of expressing a particular idea, all such manners of expression "merge" with the underlying idea. Merged expressions are not protected by copyright. This doctrine prevents any author from gaining a de facto monopoly on the underlying idea in contravention of § 102(b). See supra note 119.
substantial similarity of protected expression. Instead, Intel's strategy was to focus on discrete, isolated similarities which were largely the inherent result of the hardware similarities or of the simplicity of the sequence in question. Such evidence requires little technical understanding by the trier of fact, is relatively easy to present, and given the difficult technical subject matter, can rather easily be made to take on an exaggerated and unwarranted degree of significance. Where the trier of fact is unable to meaningfully understand two works, such a showing of similarities, whatever their significance, may be gratefully seized upon to support a finding of infringement.122

121. Because the focus of Intel's case was on whether NEC used the Intel Microcode, rather than whether the works were substantially similar, Intel also introduced evidence concerning the development history of the V20/V30, and the background of the NEC engineer who was responsible for the NEC Microcode, Hiroaki Kaneko. Intel portrayed Mr. Kaneko as young and inexperienced and argued that because he was under "tremendous time pressures", he was forced to take shortcuts in order to meet his deadlines. NEC, on the other hand, showed that Mr. Kaneko was an exceptionally talented engineer and computer scientist, that the schedule for the project was not unusual, and that the NEC Microcode was completed in a normal time frame.

Although the parties calculated development times differently, Intel did not controvert NEC's expert testimony that an engineer of Mr. Kaneko's ability could have developed the NEC Microcode in the time actually taken. Tr. IX 1429:13-1431:19. The Intel Microcode was completed and 95% debugged in less than two months. Mr. Kaneko's first draft was done in six weeks but 95% of the debugging took several more months. The "clean room" microcode, discussed infra, was completed and 95% debugged in two weeks. See infra notes 181 to 188 and accompanying text.

Intel demonstrated an impressive ability to find "copying" even in the most trivial events. Mr. Kaneko conceded at trial that it was possible he had looked at his listing of the NEC μPD8088 microcode while working on the NEC Microcode, even though he did not recall doing so. That concession was seized on by Intel's counsel as a damming admission. Tr. XXI 2964:5-8. Apparently, in Intel's eyes, looking at the Intel Microcode is the same as copying it. Intel ignored the basic principle that there is no impropriety whatsoever in reading a copyrighted work. On the contrary, the very purpose of our copyright law is to encourage authors to publish their works so that society may utilize and benefit from their teachings.

"[P]rogrammers are free to read copyrighted programs and use the ideas embodied in them in preparing their own works." Final Report, National Commission on New Technological Uses of Copyrighted Works ("CONTU") (1978) at 51. See infra note 169.

122. Intel's strategy was similar to that of other recent computer copyright plaintiffs presenting infringement claims. In practice, the computer plaintiff's strategy has been to focus on whether the defendant "copied," and pay little more than lip service to the issue of substantial similarity between the works as a whole, hoping to obfuscate this required element by convincing the Court that any "copying" constitutes infringement. M. Kramer Manufacturing Company, Inc. v. Andrews, 783 F.2d 421 (4th Cir. 1986), provides a good example of what happens when a court confuses "copying" and "substantial similarity." In Kramer, the plaintiff alleged infringement of an audiovisual copyright in a video poker game screen display. The trial court, in a bench trial, compared the two works and found they were not substantially similar in appearance. Nevertheless, on appeal, the Fourth Circuit held that the defendant's display infringed the plaintiff's work, because the defendant had "copied" a substantial part of a computer program that generated the display.
Presumably Intel hoped that the Court would overlook the fact that such similarities constituted neither similarities of protected expression nor a substantial portion of the Intel Microcode. Under different circumstances not present in this case, isolated similarities may constitute evidence of copying, the first element of the *Arnstein* test. However, the quantum and nature of substantial similarity of protected expression necessary for a finding of infringement is different. More important in the context of *NEC v. Intel*, similarities in unprotected elements, such as merged expression or scenes a faire, must be factored out of the substantial similarity analysis for the purpose of determining infringement.

Thus the district court addressed only whether the defendant's game was substantially similar to [plaintiff's] game . . . . The district court, however, lost sight of the ultimate issue: whether Drews Distributing copied the plaintiff's game. If there was clear proof of actual copying by the defendants, that is the end of the case. 783 F.2d at 445.

The occasional case such as Kramer ignores the fact that substantial similarity of protected expression is, in itself, a necessary element of plaintiff's case; it does not make any difference whether the defendant "copied", if the resulting work is not substantially similar in protected expression to the plaintiff's work. See v. Durang, 711 F.2d 141 (9th Cir. 1983) (upholding summary judgment in favor of defendant because no substantial similarity of expression between the works); *See* Cooling Systems and Flexibles, Inc. v. Stuart Radiator, Inc., 777 F.2d 485, 492 (9th Cir. 1985) ("It is true that 'courts have regarded the evidence of common errors in two similar works as the strongest evidence of piracy' [citation omitted] but proof of common errors does not obviate the need for proving substantial similarity").

This approach to computer copyright cases is discussed in detail in J. Connolly and R. Bryan, *A Unified Theory for the Litigation of Computer Software Copyright Cases*, 63 N.C.L. Rev. 563 (1985). The authors there urged that the plaintiff should not have the burden of proving the two works to be substantially similar at all. Rather, the plaintiff should only need show the accused work was "derived" from the plaintiff's work. Their approach, however, disregards the fact that even where a plaintiff accuses a work of being "derived" from plaintiff's (and thus violating plaintiff's exclusive right under § 106(2) "to prepare derivative works based upon the copyrighted work") rather than being a "copy" (in violation of plaintiff's exclusive right under § 106(1) "to reproduce the copyrighted work in copies"), the plaintiff must still show that the accused "derivative work" is itself "substantially similar" to plaintiff's work. *Litchfield v. Spielberg*, 736 F.2d 1352, 1357 (9th Cir. 1984), *cert. denied*, 105 S. Ct. 1753; *Berkic v. Creighton*, 761 F.2d 1289, 1291 (9th Cir.), *cert. denied*, 106 S. Ct. 85 (1985) ("If the plaintiff cannot show a substantial similarity between the defendants' work and his own, he cannot prevail on a claim for alleged violations of his right to prepare derivative works.")

123. *See*, e.g., *Cooling Systems and Flexibles, Inc. v. Stuart Radiator, Inc.*, 777 F.2d 485, 492 (9th Cir. 1985).


125. *See infra* notes 165 to 170 and accompanying text.

126. *See infra* note 168 and accompanying text.

127. As noted in *Warner Bros., Inc. v. American Broadcasting Companies, Inc.*, 530 F.
The nature of the two Microcodes guaranteed that Intel, after translation, would be able to show some similarities. Both the Intel Microcode and the portion of the NEC Microcode devoted to executing the 8086/8088 instruction set consist in total of about five hundred lines; the average length of the microsequences is short: about six lines. The portions of the NEC Microcode challenged by Intel are simple and even shorter than the average. Intel claimed that 54 of 88 sequences are substantially similar.128 These 54 sequences averaged in length slightly less than four lines each.129

As is discussed infra,130 NEC's position, corroborated by notes written by Intel's expert,131 was that the simpler sequences would inevitably be similar. However, virtually all of Intel's evidence of similarities related to these simpler sequences. Hence, that evidence could not show that the protected expression, if any, contained

Supp. 1187, 1190 (S.D.N.Y. 1982), aff'd 720 F.2d 231 (2d Cir. 1983), such analysis requires that courts "distill the protected parts of a work from the unprotected". See also, Cooling Systems and Flexibles, Inc. v. Stuart Radiator, Inc., 777 F.2d 485 (9th Cir. 1985).

[Plaintiff] argues that the ordinary reasonable observer would find that the catalog are virtually indistinguishable. This misses the point. What is important is . . . whether the very small amount of protectible expression in Cooling Systems catalog is substantially similar to the equivalent portions of Stuart's catalog. Id. at 492-493.

128. Intel Ex. ABZ.
129. NEC Ex. 405.
130. See infra notes 147 through 164 and accompanying text.
131. During discovery, NEC succeeded, over Intel's objection, in obtaining notes written by Intel's expert, Dr. Patterson, which became Intel Ex. R. Those notes repeatedly corroborate NEC's position, stating at various points that the expert would have to examine the longer routines in order to find any meaningful evidence of copying, because the shorter routines, being so highly constrained, must inevitably be similar.

The following are a few examples from Dr. Patterson's notes:

—"How can we tell what happened? I tried looking at the V20 microcode to see (a) if there were strange short sequences and (b) if the long sequences look as if they were copied. With emphasis on performance and limited complexity of most 8086 instructions and limited choices in similar microarchitecture, many sequences should look the same (factoring out the NEC changes to the microarchitecture)."

—"Given the similarities of the microarchitectures, for some simple 8086 instructions, there is only one efficient way to do the microroutines in both machines."

—"Argument: microarchitectures are similar, instruction sets are identical, given similarities how can you tell if there was copying or if there was good engineering on both sides? Problem: in those circumstances, good engineers would lead to virtually identical microcode."

Although during Intel's direct case, Dr. Patterson attempted to downplay the significance and dispute the accuracy of these "early impressions," he later, during Intel's rebuttal to NEC's case, finally had to concede the constraints imposed on the shorter sequences. Tr. XX 2775:4-9; 2803:14-20.
within the Intel Microcode, when taken as a whole, was substantially similar to the NEC Microcode. Moreover, simply on a common sense level, Intel never explained why NEC would copy the shorter, simpler sequences, but not the longer sequences which are more difficult to create independently.

Although Intel's expert testified that the 54 sequences were substantially similar, he seemed to equate that term with any similarity, whether inherent or not. This occasionally led to bizarre results. For example, Dr. Patterson was able on the same sequence to find the NEC version "substantially similar" to Intel's, while at the same time observing that the NEC routine was "quite different" from the Intel routine and "not like [the] 8086 at all." This strange logic which can make the same microcode both quite different and substantially similar is discussed below.

A few illustrations will show the nature of the similarities relied on by Intel in lieu of substantial similarity of protected expression. For example, there is a sequence, "RESET", in which six specific registers are initialized (that is, set to a specific starting value, either a zero or a one). The only choice available to the microcode designer is the order in which the registers are initialized; moreover, the theoretically possible orders emphasized by Intel are in reality significantly restricted by the functioning of the hardware. The two RESET sequences, after being translated into a common language, are as follows:

<table>
<thead>
<tr>
<th>NEC</th>
<th>INTEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSO=0</td>
<td>DSO=0</td>
</tr>
<tr>
<td>b=0</td>
<td>QHLD</td>
</tr>
<tr>
<td>flag=0</td>
<td>PS=bus</td>
</tr>
<tr>
<td>PS=bus</td>
<td>PC=0</td>
</tr>
<tr>
<td>PC=0</td>
<td>QINI</td>
</tr>
<tr>
<td>DS1=0</td>
<td>flag=0</td>
</tr>
<tr>
<td>SS=0</td>
<td>DS1=0</td>
</tr>
<tr>
<td></td>
<td>SS=0</td>
</tr>
<tr>
<td></td>
<td>RNI</td>
</tr>
</tbody>
</table>

Without considerable background and explanation, a trier of fact could not rationally decide whether the NEC sequence is substantially similar in protected expression to the Intel sequence. To decide that question one must understand the sequences, the other

---

132. LOOPZ, Intel Ex. ABZ.
135. See infra text accompanying notes 138-140.
136. Tr. XII 1903:25-1907:11.
possible orders in which the instructions might be performed and, most importantly, what effect architectural constraints impose on alternative RESETs, or, to state it another way, to what degree a noncopied RESET might appear different. Yet, Intel made little effort to show that a noncopied RESET would be materially different from NEC's. Instead, Intel argued that NEC, in an earlier unused version of its Microcode, had copied Intel's RESET sequence. Because the version NEC used was "derived" from that earlier copy, Intel argued, it too is a "copy" and therefore "substantially similar".

At times, the similarities found by Intel's expert completely ignored the actual NEC Microcode. For example, there is a very simple sequence in both the NEC and Intel Microcodes that generates "1" and puts it in a temporary register. Even though the operation is simple, however, the NEC and Intel Microcodes perform the operation differently. NEC generates the "1" directly. Intel generates

---

137. As pointed out in Cooling Systems, two works may be virtually indistinguishable to the "ordinary reasonable observer" without being "substantially similar" if the similarities are primarily unprotectible or merged expression. Cooling Systems & Flexibles, Inc. v. Stuart Radiator, Inc., 777 F.2d at 492-493 (9th Cir. 1985).

138. Intel Amended Proposed Finding No. 31(a). Intel never explained precisely what NEC copied. Apparently the order in which the registers are initialized is the claimed expression, but if so, NEC's order is different.

This approach by Intel of looking to earlier, different versions of the NEC Microcode to show "substantial similarity" between the NEC and Intel Microcodes, in addition to its inherent absurdity, is plainly contrary to well-established law.

"Plaintiff originally listed 18 points of similarity, based on a comparison of his script with the defendant's original two scripts. Seven of the allegedly similar scenes or phrases were, however, not incorporated in the film as shown on television. Since the ultimate test of infringement must be the television film as produced and broadcast, not the preliminary scripts, we have confined our analysis to the remaining eleven allegedly similar scenes which were in the film as produced and broadcast." Fuld v. National Broadcasting Co., 390 F. Supp. 877, 882, n.4, (S.D.N.Y., 1975). See also, Davis v. United Artists, Inc., 547 F. Supp. 722, 724, n.9, (S.D.N.Y., 1982).

The issue is whether the final version of the allegedly infringing work, the version distributed to the American public, is substantially similar to the copyrighted work. Because no amount of similarity in an earlier version can make the work in issue more or less similar to the copyrighted work, earlier versions are irrelevant to this issue.

The Ninth Circuit has taken this rule one step further. In See v. Durang, 711 F.2d 141 (9th Cir. 1983), the trial court had granted summary judgment for defendant because it found, as a matter of law, a lack of substantial similarity. Plaintiff argued that this ruling had precluded him from discovering earlier drafts of defendant's work, which plaintiff argued might have reflected copying from plaintiff's work which was disguised or deleted in later drafts. In upholding the summary judgment and thus preventing plaintiff from discovering defendant's earlier drafts, the Ninth Circuit ruled that "copying deleted or so disguised as to be unrecognizable is not copying." Id. at 142. Thus, under the Ninth Circuit view, earlier versions are not merely irrelevant to the issue of substantial similarity, but they are not even properly discoverable on that issue.
it indirectly. After translation by Intel, the companies' respective sequences appeared as follows:

<table>
<thead>
<tr>
<th>NEC</th>
<th>INTEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 → TMPBL;</td>
<td>CR → OPR; Unc(5);</td>
</tr>
<tr>
<td>Susp;</td>
<td></td>
</tr>
<tr>
<td>Jump (02)</td>
<td></td>
</tr>
</tbody>
</table>

Despite the obvious differences, Intel's expert concluded that NEC used the Intel sequence in developing the NEC sequence. Thus, even though it was different, because in his view it was copied, it became substantially similar.139

Under Intel's theory, it is hard to imagine how NEC's sequence could be made noninfringing. The "1" is required. NEC's sequence is different from Intel's in the only substantial way possible: the only operation performed, generating "1", is done in a different way. Because Intel's expert thought Mr. Kaneko used the Intel Microcode to produce different microcode, the resulting microcode automatically became substantially similar no matter how different it was in reality.140

Another of Intel's claimed "smoking guns" clearly shows the difference between similarities of the sort found by Intel and the substantial similarity in protected expression required for a finding of infringement. There is only one place in both the V20/V30 and 8086/8088 where the microcode differs between the 8-bit and 16-bit versions because of a difference in the respective chip designs. Intel claimed that, had a different hardware design been chosen, the same block of microcode could have been created to work on both the 8-bit and 16-bit versions of the parties' respective devices. Therefore, merely because the same microcode could not be used, Intel labeled the design a "hardware bug" which was contained in both the V20 and 8088. The difference in microcode required by the 8088 hardware was thereupon characterized by Intel's expert as a "patch" around this bug. NEC had adopted the same hardware design, and, of necessity, had the same difference in microcode between its 8-bit and 16-bit devices (i.e., the same "patch"). To Intel's expert, this

140. Intel's expert had inadvertently translated the wrong NEC code, that is, an intermediate version, rather than the allegedly infringing work. The sequence in the version actually used by NEC in the V20/V30 and which appears in the text above, is less similar to the Intel sequence than NEC's earlier unused version had been. NEC Ex. 179A. But to Intel, such a change did not matter. Once a sequence was classified as "substantially similar" (i.e., Dr. Patterson thought it was "copied") changes could not make it less "substantially similar". This approach, as noted above, is contrary to well-settled law. See supra note 138.
was convincing evidence of copying, apparently on the theory, not supported by the facts, that NEC designed the hardware containing the bug only after the microcode had been designed. However, all relevant witnesses, including both parties' experts, as well as the creator of the Intel "patch", agreed that, given the hardware, the microcode employed by both companies was reasonable. Any good engineer would produce similar microcode in similar circumstances.

Intel did not limit its search for similarities which it could urge evidenced copying to a comparison of individual sequences. For example, Intel also argued that the existence of some similar groupings by the Intel and the NEC microcode designers of more than one macroinstruction in a single microsequence constituted evidence of copying. However, as both NEC's expert and an Intel engineer testified, it would normally be expected that some of the groupings would be the same, while others would be different, simply as a result of good engineering. Even Intel's expert agreed that if he had created the NEC Microcode, some of the challenged groupings would probably have been the same.

141. This theory was contrary to the facts. A full enumeration of the factual errors and faulty assumptions that led to Dr. Patterson's conclusion is beyond the scope of this article. It is sufficient to note that Intel confused circuit design with hardware specification. Mr. Kaneko did not have to know every detail of the final hardware design to produce the NEC Microcode. Tr. III 461:4-15. The NEC hardware had already been specified by Mr. Kaneko so he knew full well how the hardware would behave, long before circuit design was completed. That is all a microcode designer needs.


143. For example, the "clean room" microcode, discussed infra, contains the same "patch", to no one's surprise. See infra notes 180 to 187 and accompanying text. Even Dr. Patterson conceded that there is "a good reason for the code to appear the way it did." Tr. XX 2810:20-21.

144. Tr. IX 1399:2-25.

145. Tr. XX 2794:8-14. Interestingly, the percentage of groupings in the 80186 microcode section of the V-Series which are similar to the groupings in Intel's 80186 microcode is greater than in the 8086 section. NEC Ex. 375; Tr. IX 1400:1-1402:9. Yet, Intel has never claimed that NEC copied the 80186 microcode and indeed NEC had no access to it.

Another of Intel's more unusual copying theories also involved groupings. NEC, in an earlier version of its Microcode, had grouped its byte/word sequences differently from Intel to achieve better performance. Later, because of severe space constraints, Mr. Kaneko had to forego that performance to save space and the microcodes became more similar. From those facts, Intel's counsel hypothesized as follows:

Mr. Kaneko did start out to write faster code through different groupings. His [first draft] microcode did have some different groupings and, as a result, it gained some speed . . . .

[Later] Mr. Kaneko was instructed to cut back the use of space for the 8086
Thus, Intel's direct case consisted of discussing a handful of sequences in which its expert felt some similarity existed, or had once existed, between the two Microcodes. NEC disputed whether there was anything at all unusual in those similarities, or, in some instances, whether there were similarities at all. Of greatest importance, however, Intel made no effort, beyond the bare assertion that 54 sequences were substantially similar, to explain what protected expression it claimed NEC had appropriated.

4. The Absence of Substantial Similarity

NEC's response to Intel's copying claims was twofold. First, NEC showed that, contrary to Intel's assertions, the NEC Microcode was developed independently of the Intel Microcode. That evidence consisted primarily of the V20/V30 development history, showing that NEC revised its Microcode repeatedly over a two year period, during which time many approaches were tried and rejected before the NEC Microcode was finalized. Second, NEC showed that those similarities which did exist between the NEC and Intel Microcodes were inevitable given the similar microarchitectures and same macroinstruction sets and so could not constitute the substantial similarity of protected expression which is necessary to a finding of infringement. The latter point, involving the merger doctrine, raised the most difficult questions at trial. The difficulty arose because, unlike Intel's fragmented evidence of similarities, a reasoned substantial similarity analysis re-

---

... instruction, and he did that by adopting the twelve out of thirteen [like Intel]. It was biased by knowledge that this is how Intel achieved its similar space objective... we know all about what Intel did. You spend all that time studying it and you do the same thing, and the chances of doing it by coincidence is pretty slim...

Tr. XXI 2939:4-2941:18.

Apparentiy, because Mr. Kaneko "knew" what decisions Intel had made, he was not allowed to make the same decisions. Intel's counsel even referred to Mr. Kaneko's knowledge of the µPD8086 as a form of "disability". Tr. XXI 3079:7-14. Intel's counsel seemed to have overlooked the basic axiom that programmers are allowed to look at and learn from each other's work. See supra note 121.

146. In some instances, the similarity relied upon by Intel had disappeared from the NEC Microcode during the lengthy development process. See supra notes 136 to 138 and accompanying text.

147. Development of the NEC Microcode took more than two years, measured from the date work began, to the date the final preproduction changes were made, and a half-dozen different versions were introduced at trial. For example, RESET, the sequence noted above, was revised by NEC at least four times; see supra notes 136 to 138 and accompanying text.


149. See infra notes 165 to 170 and accompanying text.
quires some understanding of microcode and its relationship to the microprocessor of which it forms a part.\textsuperscript{150}

Microcode is not created in a vacuum, as one might write a novel. Microcode performs a function, that of executing a set of macroinstructions on a particular given microarchitecture or hardware, and so is highly dependent on (i.e., constrained by) both of those elements.

The 8086/8088 macroinstructions are published in the Intel user's manual and allow programmers to write application programs for use on the microprocessors. The V20 and V30 were intended, among other things, to be able to act as enhanced substitutes for the Intel 8086 and 8088. As such, the macroinstruction set of the V-Series includes, as a subset, the macroinstruction set of the 8086/8088. Intel has not claimed any copyright in its macroinstruction set, which is in the public domain, and acknowledges that NEC, like anyone else, is free to use those macroinstructions.\textsuperscript{151}

Similarly, NEC used a microarchitecture which had many similarities to Intel's, but which also includes significant additions, including a dual bus, an effective address generator, a shifter and loop counter, and random logic multiply and divide, all of which allow the V-Series to outperform the 8086/8088. Intel has also recognized NEC's right to use Intel's microarchitecture,\textsuperscript{152} which can be protected, if at all, only under the patent laws\textsuperscript{153} and as mentioned above, the parties are licensed under each other's patents.\textsuperscript{154}

At the core of the NEC/Intel dispute is the amount of creative freedom available to a microcode designer. Microcode is heavily dependent on both the macroinstruction set and the microarchitecture. Thus, a microcode designer is highly constrained in the choices available to him in developing microcode for a particular device. For many of the microsequences, which are generally the shorter, simpler sequences, the available choices are very limited, often amounting to no more than trivial variations. For example, as

\textsuperscript{150} Intel rested its direct case on the fifth day of the twenty-one day trial, having made little effort to make the technology involved understandable to the Court. That burden thus fell on NEC by default because a well-reasoned substantial similarity analysis requires such understanding.

\textsuperscript{151} NEC Corp. v. Intel, 645 F. Supp. 590, 592 (N.D. Cal. 1986), Partial Finding of Fact No. 15C.

\textsuperscript{152} \textit{Id.}, Partial Finding of Fact No. 15D.

\textsuperscript{153} \textit{See supra} note 22.

\textsuperscript{154} \textit{But see infra} notes 171 through 179 and \textit{accompanying text} regarding Intel's "bootsrap license" argument whereby Intel seeks to nullify this acknowledged right of NEC to use Intel's patents by an improper extension of the copyright law.
was pointed out by NEC's expert, Dr. Gideon Frieder, there are only a limited number of ways in which one can add two numbers.

In contrast, as one moves to sequences which are longer and more complicated, the choices available to the microcode designer become correspondingly greater. Consequently, given that the NEC and Intel microcode designers were both working under substantially the same constraints, one would expect a fair degree of similarity in function between the two Microcodes, even in the absence of copying, especially in the shorter, simpler microsequences. This is precisely the situation which was shown at the trial.

At the outset, it must be noted that much of the NEC Microcode was not even alleged to be similar. The 8086/8088 instruction set consists of over 250 possible machine instructions. These instructions are implemented in the 8086/8088 through 88 documented microsequences in the Intel Microcode. At trial, Intel asserted that only 54 of 88 sequences were substantially similar. In terms of total lines, these amounted to only about forty percent of the microcode implementing the 8086/8088 instruction set. Thus, before trial began, Intel had conceded that sixty percent of the NEC Microcode was not similar.

NEC showed, however, that the allegedly similar forty percent contained for the most part, short simple routines where, due to the physical constraints, there was usually little or no choice available to the microcode designer. On the other hand, the portion which Intel conceded was different included all of the longer, more complex routines. The average length of those sequences not on Intel's list of substantially similar sequences is more than double the average length of the sequences which were alleged to be substantially similar. The allegedly similar sequences averaged less than four lines in length.

NEC demonstrated these facts not by bare assertions by its expert, but by an extensive excursion into the world of the microcode designer. NEC first presented to Judge Ingram a comprehensive tutorial on what microcode is and how it works in the 8086/8088 and V-Series, followed by a painstaking analysis of the individual

155. See text at note 107 supra regarding the undocumented microsequence hidden by Intel in the Intel Microcode which executes the "magic instruction."

156. NEC Ex. 375. Intel never explained why NEC would "copy" the simplest, easiest routines, and not copy the longer, harder sequences. For example, an Intel engineer testified that different versions of RESET could be created in a matter of minutes. Tr. III 540:22-541:8. Of course, even the more complex sequences were not extraordinary. The same Intel engineer testified "the algorithms used were very straightforward algorithms. They have been around since the fifties." Tr. IV 557:10-12.
sequences on Intel's list of 54, showing the constraints on NEC which dictated the similarities.

The vast majority of these microcode sequences are four or fewer lines long. For example, there are machine instructions called "TEST eb, rb" and "TEST ew, rw" which are implemented by the following microsequence:

<table>
<thead>
<tr>
<th>NEC</th>
<th>INTEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>a=RM; b=R; AND(a); ENDN</td>
<td>P(M, TMPA, 1, LAND, TA, O);</td>
</tr>
<tr>
<td>bus=ALU;</td>
<td>P(N, TMPB, 4, XX, NXT, O);</td>
</tr>
<tr>
<td></td>
<td>P(S, XX, 4, XX, RN1, 1);</td>
</tr>
</tbody>
</table>

Intel's first step, of course, was to translate both Microcodes into a hybrid common language thereby eliminating the differences between the two Microcodes reflecting the NEC dual bus (a significant difference between the two machines). Intel's translation of TEST resulted in the following common language microcode:  

<table>
<thead>
<tr>
<th>NEC</th>
<th>INTEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>M → TMPA;</td>
<td>M → TMPA; LAND(TA);</td>
</tr>
<tr>
<td>N → TMPB;</td>
<td>N → TMPB; NXT;</td>
</tr>
<tr>
<td>S → XX; F;</td>
<td>S → XX; F; RNI;</td>
</tr>
</tbody>
</table>

As one can see, when translated the two sequences are made to appear very similar. On the other hand, given the simple nature of the TEST operation, it would be surprising if the codes were not similar in the way they function. Because of the hardware used, the only option or choice available to either Intel or NEC was which temporary register ("TMPA" or "TMPB") to use. To Intel, that was no choice at all. Intel's expert testified that in deciding whether sequences were "substantially similar", the "selection of particular temporary registers is not one of the things that influences my conclusions." In short, in TEST, Intel had factored out of its similarity analysis the only choice available to NEC, so that

157. NEC objects to the "translations" performed by Intel. See supra notes 106 to 110 and accompanying text.

158. The TEST instruction, as defined in the Intel manual, "performs the logical 'and' [denoted in Intel's common language as "LAND(TA)"] of the two operands (byte or word) [denoted in Intel's common language as "M" and "N"], updates the flags [denoted in Intel's common language as "S" and "F"], but does not return the result [denoted in Intel's common language as "XX"]. . . ." Initially, the contents of "M" and "N" had to be moved to temporary registers ["M → TMPA" and "N → TMPB", as denoted in Intel's common language] in order to perform a logical "and".

159. Tr. III 426:15-18.
no matter what NEC did Intel could claim the sequence was substantially similar.

In response to NEC’s evidence that the similarities were inevitable, Intel first attempted to show that many alternatives existed for several of the sequences which NEC claimed were constrained by the hardware. The most elaborate and perhaps the most dramatic of these attempts was a list of over two hundred “alternatives” generated by Intel engineers for one of the NEC routines. Upon further analysis, however, it was shown that twenty-four of those alternatives would in fact not work on the V-Series, approximately one hundred and seventy were unreasonable and unprofessional, while only about eight were in fact reasonable alternatives. However, each of those eight included only trivial variations of the NEC sequence, involving mostly differences in the choice of temporary registers. Accordingly, Intel’s attack on constraints resulted in alternatives which were either nonfunctional, unreasonable, or trivial variations of the actual NEC sequence. Intel’s alternatives themselves became highly probative evidence that NEC often had little or no meaningful choice in the way in which a microsequence could be created.

160. MOV mem acc. Intel Ex. AGM.
161. NEC Ex. 427.
162. These unreasonable “alternatives” involved steps such as, while moving information from one register to another, passing that information through the ALU and adding zero to it. Even the creator of the Intel Microcode (called as a witness for Intel) testified that such an operation was an error. Tr. III 483:17-20. This type of alternative would be akin to suggesting, as an alternate way of traveling by car from San Jose to San Francisco, that the driver exit the freeway at various points, circle a few side streets, and stop and restart the car before continuing on the freeway from where he had exited. Such a circuitous route is not a reasonable alternative. As stated by the court in Kepner-Tregoe, Inc. v. Carabio, 203 U.S.P.Q. (BNA) 124, 131 (E.D. Mich. 1979):

To cite an extreme, how many ways may one describe how to draw a circle with a compass? To be sure, many outlandish methods using eccentric styles could be conceived. But there is no societal value in that. For better it is to narrow the protection to the first writer, in the interest of competition.

See also Cooling Systems and Flexibles, Inc. v. Stuart Radiator, Inc., 777 F.2d 485, 492 (9th Cir. 1985) (“Surely Cooling Systems cannot contend that Stuart, to escape a charge of copyright infringement, must restrict itself to producing a catalog not usable by those for whom it is intended.”) Clearly, Intel cannot force NEC to utilize outlandish microcode simply to be different.

163. Dr. Patterson attempted to justify Intel’s circuitous alternatives to another sequence by positing that a microcode designer might add extra operations in the middle of a sequence merely “to exercise the hardware.” Tr. XX 2812:8-12.

164. One side effect of Intel’s efforts to develop alternative microcode was to illustrate how different the V20 and 8088 actually are. Although Intel’s engineers were presumably experienced with the Intel Microcode, they had a very difficult time developing debugged NEC microcode. The longest NEC sequence Intel attempted to redesign was 14 lines, and contained so many bugs the effort was abandoned. Intel Ex. ACZ; Tr. XIII 2010:8-2011:12.
In sum, the evidence at trial showed that the microcode designers often had few choices available. Where choices were available, sometimes NEC made the same choice as Intel, and sometimes made a different choice. Indeed, none of the 54 NEC routines on Intel's list of substantially similar sequences is identical, even after translation by Intel, to the corresponding Intel routine. Where a greater number of choices were available to the NEC microcode designer (i.e., in the longer, more complex routines) the sequences were so different that Intel did not even allege similarities in these sequences. This is all as one would expect of microcode developed independently under the constraints imposed by the Intel macroinstruction set and similar microarchitectures.

5. The Legal Significance of Inevitable Similarities

That a microcode designer is often highly constrained in his choices by the macroinstruction set and the microarchitecture is relevant to a copyright analysis in two respects. The first is the result of a legal doctrine known as the merger doctrine. The second has to do with the evidentiary weight to be accorded similarities between two works.

The copyright law does not grant an author a monopoly over the ideas expressed in his work; rather, only the author's particular expression of those ideas is protected against copying.\textsuperscript{165} The idea expressed by both the NEC Microcode and the Intel Microcode is the execution of the 8086/8088 macroinstruction set on similar microarchitecture.\textsuperscript{166}

In cases where there are but one or a limited number of ways of expressing a given idea, protecting the expression of that idea would create de facto protection of the underlying idea. In this situation,


\textsuperscript{166} Intel did not agree that the unprotected idea included the microarchitecture. But what is the microarchitecture if not "idea"? One cannot select microcode to move data from "Register A" to "Register B" without first deciding that there will be two registers, "A" and "B" and that there will be some means of moving data between them. Those elements of the microarchitecture are certainly not part of the microcode's expression; rather they are part of the idea being expressed. Before the Intel microcode designer could express the idea (i.e., operation) of moving data between "A" and "B" he must first have in mind the idea of those registers and data paths, and surely NEC, in addressing the same functions, is entitled to have the same idea in mind.

"One is always free to make the machine do the same thing as it would if it had the copyrighted work placed in it, but only by one's own creative effort rather than by piracy." SAS Institute, Inc. v. S & H Computer Systems, Inc., 605 F. Supp. 816, 829 (M.D. Tenn. 1985), quoting from the CONTU Report. See infra note 169.
under the merger doctrine, the idea and the expression of the idea are said to merge. Copyright protection does not extend to this merged expression because such protection would afford the copyright owner a monopoly over the underlying idea.\textsuperscript{167} Thus, anyone can freely use the merged expression. Moreover, under the "scenes a faire" doctrine, "similarity of expression . . . which necessarily results from the fact that the common idea is only capable of expression in more or less stereotyped form will preclude a finding of actionable similarity."\textsuperscript{168} In the NEC \textit{v. Intel} case, to the extent that the idea of interpreting macroinstructions on a particular sort of computer architecture can be expressed (in microcode) in only limited ways, or in a more or less stereotyped form, all such expressions merge with the underlying idea.\textsuperscript{169} Quite apart from the

\textsuperscript{167} Herbert Rosenthal Jewelry Corp. \textit{v. Kalpakian}, 446 F.2d 738, 742 (9th Cir. 1970).

\textsuperscript{168} The copyright law protects expression of unprotectible ideas only insofar as is possible without protecting the ideas themselves.” Landsberg \textit{v. Scrabble Crossword Game Players, Inc.}, 736 F.2d 485, 489 (9th Cir. 1984), cert. \textit{denied}, 105 S. Ct. 513 (1984). An insightful analysis of a substantial similarity/merger argument identical to that raised by NEC was provided by the Second Circuit in a case involving rulebooks for a tabletop soccer game. Affiliated Hospital Products, Inc. \textit{v. Merdel Game Manufacturing Company}, 513 F.2d 1183 (2d Cir. 1975).

\textsuperscript{169} The court analyzed this issue on facts remarkably analogous to those in this case:
The issue that is squarely raised, therefore, is to what extent a copyright holder can prevent a competitor from publishing a similar rulebook to the copyrighted rulebook. No claim is of course made that appellant can protect the game of Caroms or its variations which are in the public domain. The rules of the game are perforce in the public domain as well as the game itself. Affiliated's copyright only protects Affiliated's arrangement of the rules and the manner of their presentation, and not their content. Here, however, the simplicity of the games makes the subject matter extremely narrow, and the distinction between substance and arrangement blurs. On these facts we hold that Merdel, although admitting to access and use of Affiliated's work, did not infringe Affiliated's copyright. Merdel did not copy Affiliated's rules verbatim, and indeed its changes enhanced the clarity of the rules. This conclusion, based on the facts here present, in no way affects the general rule, applicable in other situations, that upon a showing of access to copyrighted material, an alleged infringer cannot escape liability for his appropriation through the introduction of slight changes. [citation omitted] We are encouraged in this conclusion through recognition of the fact that a contrary result would prevent publication of the rules of any simple game in the public domain unless the second entrant in the field developed his rules solely through watching the game being played; a result which would afford protection to the game itself.

\textit{Id.} at 1188-89.


\textsuperscript{169} The National Commission on New Technological Uses of Copyrighted Works ("CONTU") was created by Congress in 1974 as a part of the effort to revise comprehensively the Copyright Laws. P.L. 93-573, 88 Stat. 1873. Part of CONTU's charge was to consider to what extent computer programs should be protected by copyright. CONTU issued its Final Report in 1978 in which it recommended the continued copyrightability of
merger doctrine, constraints limit the evidentiary weight to be accorded similarities; no inference of copying can be drawn from inevitable similarities.

These considerations are quite important to the NEC v. Intel case. Because the creator of the Intel Microcode was under severe constraints imposed by the macroinstruction set and the microarchitecture, the scope of copyright protection, if any, to be afforded the Intel Microcode is very narrow and a near identity between the two Microcodes must be shown before infringement can be found.170

6. Intel’s “Boot Strap License” Argument

Between Intel’s expert’s concession that the microarchitecture offers limited choices,171 and Intel’s failure to produce dissimilar172 reasonable alternatives to the Intel sequences,173 Intel’s response to the merger doctrine gradually took a different shape. Ultimately Intel tacitly admitted that a high degree of similarity in those sequences constrained by the architecture was inevitable, and shifted the emphasis of its case to an argument, novel in intellectual property law, that the constraints imposed on NEC should be disre-

---

computer programs and also recommended appropriate changes in 17 U.S.C §§ 101 and 117, which changes were adopted by Congress verbatim. See, Apple Computer, Inc. v. Formula International, Inc., 725 F.2d 521, 524-525 (9th Cir. 1984). The CONTU Report recognized the applicability of the merger doctrine to computer programs.

The “idea-expression identity” exception provides that copyrighted language may be copied without infringing when there is but a limited number of ways to express a given idea. This rule is the logical extension of the fundamental principle that copyright cannot protect ideas. In the computer context this means that when specific instructions, even though previously copyrighted, are the only and essential means of accomplishing a given task, their later use by another will not amount to an infringement.

. . . copyright protection for programs does not threaten to block the use of ideas or of program language previously developed by others when that use is necessary to achieve a certain result. When other language is available, programmers are free to read copyrighted programs and use the ideas embodied in them in preparing their own works. This practice, of course, is impossible under a patent system, where the process itself is protected, and difficult under trade secrecy, where the text of a program is designed not to be revealed.

CONTU Report at pp. 50-51.

170. “Where idea and expression coincide, there will be protection against nothing other than identical copying of the work.” Sid & Marty Krofft Television Productions, Inc. v. McDonald’s Corp., 562 F.2d 1157, 1168 (9th Cir. 1977).

171. See supra note 131.

172. In closing argument, Intel urged the Court to ignore the fact that its alternatives were similar: “Who cares if they are similar? . . . And the fact that some of these, thousands of variables in terms of rewrites, appear similar to each other, sure; so what? They are still different from each other, too. That is the point.” Tr. XXI 3071:23-3072:9.

173. See supra notes 160 to 164 and accompanying text.
garded because they were self-imposed, that is, the result of NEC's decision to employ a microarchitecture which was similar to that of Intel. In other words, if similar hardware would necessarily result in similar microcode, then, Intel claimed, NEC should not be permitted to use similar hardware. Intel advanced this argument even though Intel agreed that NEC had a right to employ similar microarchitecture, and indeed had Intel's permission to so so under the Patent Cross-License Agreement.

This became Intel's bootstrap license argument that NEC should not be allowed to "bootstrap" its right to use hardware similar to Intel's hardware into a right to use microcode sequences which are similar to those of Intel. Rather, NEC should have chosen different hardware, which would have necessitated different microcode. As Intel's counsel stated in closing argument:

The point that we are making . . . is that if they had just done any of these things differently, including made the architecture differently, Mr. Kaneko would have been liberated and he wouldn't have had to have all of these, the microprograms that were substantially similar. Quite to the contrary. They would have been different, and that is exactly the point. 174

This line of argument was taken to its irrational extreme by Dr. Patterson when he testified for Intel:

If we look at the first system specifications, we can see a lot of similarities in the microarchitecture to the 8088. Now, Dr. Frieder, in his analysis [just assumed] that what happened at the end was very many of the routines were very similar just as a matter of coincidence, because they were constrained. In my view, it is that [NEC] started at the disassembled code, provided a microarchitecture that would allow them to refer to this disassembled code, and copy the microprogram. 175

Dr. Patterson's hypothesis ignores reality. If NEC had started with the Intel Microcode and set out to design a microarchitecture which would allow NEC to use the Intel Microcode, surely NEC

---

174. Tr. XXI 3069:6-14. Note, however, that claiming different hardware will produce different microcode is not the same as claiming different hardware will produce noninfringing microcode. As noted supra, a “translation” can eliminate or minimize hardware differences. See notes 106 to 110 and accompanying text. The NEC design has a number of differences that Intel either disregarded in its translation (for example, the dual bus), or ignored in its similarity analysis. Intel also ignored differences required by the hardware in deciding whether two sequences were similar. Tr. XX 2836:10-17. If one adopts translation rules to eliminate the hardware differences, any microcode implementing the 8086 instruction set can be made to appear similar. A translation was simply easier for Intel to perform in this case, because the microarchitectures were fairly similar.

175. Tr. XX 2843:16-2844:10.
would have copied the longer, more complex sequences, the very areas which are most difficult to create. Moreover, Dr. Patterson himself stated in his notes that one must look to the longer sequences to find meaningful evidence of copying. Yet Intel admitted that the longer sequences, to which one would look to detect copying, were not similar at all.

Intel's bootstrap license argument also confuses the nature of the merger doctrine. The expression that is merged is Intel's expression. Nothing NEC does can alter the scope of protection available to Intel. The constraints at issue, for purposes of merger, were the constraints chosen by Intel, not NEC. The microarchitecture used by NEC was relevant to merger only to the extent it happened to be similar to that used by Intel.

The argument that, by some mysterious process, copyright law prohibited NEC from using some of the same hardware or architecture as Intel, even though NEC was licensed under Intel's patents to use that hardware, comes closer to what this case is really about: NEC has used (as it was entitled to do) a similar microarchitecture. That was the alleged copying that actually concerned Intel. Microcode can be developed afresh in a matter of a few weeks. However, securing a de facto monopoly over the microprocessor design via copyright law, without meeting the rigorous requirements of patent law, would be of great value to Intel. It would extend copyright protection to the entire microprocessor hardware architecture, without the safeguards associated with the patent system, thereby dragging computer architecture into the black hole of software copyright.

176. See supra note 131.
177. Of course the constraints imposed on NEC are relevant to another issue: whether NEC copied. The constraints on NEC served to rebut any inference that similarities were due to copying. For example, there were places where NEC made "space/performance" tradeoffs similar to those made by Intel. NEC showed those similarities were the result of constraints imposed by the NEC design, not copying.
178. See supra note 121.
179. See Copyright Protection: Legal Black Hole? Vol. 3, No. 8 COMPUTER LAW STRATEGIST 3 (Dec. 1986), wherein it is observed that Judge Ingram's ruling that the Intel Microcode is copyrightable, in combination with the recent decisions in Broderbund Software, Inc. v. Unison World, Inc., 648 F. Supp. 1127 (N.D. Cal. 1986) and Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc., 609 F. Supp. 1307 (E.D. Pa. 1985), aff'd, 797 F.2d 1222 (3d Cir. 1986), "may define copyright protection into a legal black hole, sucking in all aspects of a computer system, from the lowest microcode inseparable from the hardware, to the 'look and feel' of the user interface." However, perhaps such pessimism is yet premature, as witnessed, for example, by the Fifth Circuit's recent rejection of Whelan in Plains Cotton Cooperative Association of Lubbock, Texas v. Goodpasture Computer Service, Inc., 807 F.2d 1256, 1262 (5th Cir. 1987).
7. NEC's Clean Room Microcode

Copyright law only forbids the copying of another's work, not one's own independent development of a work even if that work turns out to be identical to some earlier copyrighted work. If there is no access to the plaintiff's protected expression, there can be no copying, and thus there can be no infringement. In the copyright context, a clean room refers to an environment where the creator of a new competitive work has no access to the protected expression contained in the competitor's work, although the creator can have access to the unprotected ideas contained in that competitor's work.

NEC, as a matter of business caution, decided prior to the start of trial to use such a clean room to produce a new microcode for the V-Series. A substitute microcode (the Clean Room Microcode) was developed within a relatively short time. This new Clean Room Microcode, developed by an outside consultant who was shielded from all access to the Intel Microcode, has virtually the same degree of similarity to the Intel Microcode as does the allegedly infringing NEC Microcode. The same similarities, with only minor variations, that Intel urged proved copying also exist in the Clean Room Microcode. This is hardly surprising since these similarities were, as was urged by NEC at trial, the result of the very limited number of choices available to the microcode designer, rather than, as urged by Intel, evidence of copying.

The Clean Room Microcode is most similar to the NEC and Intel Microcodes, as one would have predicted, in the short, simple sequences. For example, the RESETs (after translation) compare as follows:

---


181. The concept of a "clean room," of course, is suggested by the phrase originally used to describe a room used in the semiconductor manufacturing process in which outside contaminants such as dust particles were excluded to ensure the purity of the silicon being processed within the room.

182. Development of the Clean Room Microcode took far less time than the trial itself.

183. See supra notes 136 to 138 and accompanying text.
Clean Room | NEC | Intel
---|---|---
DSO=0 | QHLD | DSO=0 | NOP(A) | DSO=0 | QHLD
Flag=0 | NOP(A) | Flag=0 | QHLD | CS=bus | Flag=0
DS1=0 | MFS | CS=bus | PC=0 | QINI | CS=bus
CS=bus | PC=0 | QINI | PC=0 | QINI | PC=0
PC=0 | ENDN QINI | DS1=0 | ENDN MFS | DS1=0 | RNI
SS=0 | SS=0 | SS=0 | SS=0 | RNI

TEST\textsuperscript{184} (after using Intel translation rules) provides another example:

<table>
<thead>
<tr>
<th>Clean Room</th>
<th>NEC</th>
<th>Intel</th>
</tr>
</thead>
<tbody>
<tr>
<td>M → TMPB; LAND(TA);</td>
<td>M → TMPA; LAND(TA);</td>
<td>M → TMPA; LAND(TA)</td>
</tr>
<tr>
<td>N → TMPA; ENDN;</td>
<td>N → TMPB; ENDN;</td>
<td>N → TMPB; NXT;</td>
</tr>
<tr>
<td>S → XX; F;</td>
<td>S → XX; F;</td>
<td>S → XX; F; RNI;</td>
</tr>
</tbody>
</table>

The sequences are nearly identical. The only difference is in the selection of temporary registers which is basically a fifty/fifty chance.

How does the clean room bear on the case? First, it will bring the issue of substantial similarity back into focus, and will provide a benchmark to show how non-substantially similar microcode would appear.\textsuperscript{185} Second, it will refute once and for all Intel's claims concerning the kind of similarities that show copying by Mr. Kaneko.\textsuperscript{186} Third, it will bring to the fore the issue of architectural similarities. The Microcodes are functionally similar because the microarchitectures are similar and the macroinstruction sets are the same.

What may then be at stake when this litigation resumes is the entire notion of a clean room, and ultimately the limit to which a microcode copyright can protect hardware design. If NEC prevails, the scope of copyright protection, if any, for a work such as the Intel Microcode, will be narrow, so as to protect against only works that are similar in unexpected or unusual ways, i.e. where the creator has true freedom of choice. On the other hand, if Intel prevails, the scope of a microcode copyright will be very broad, encompass-

\textsuperscript{184} See supra notes 138 to 140 and accompanying text.

\textsuperscript{185} Similarities which are the consequence of logical coding cannot constitute protected expression. If two engineers working independently would likely produce very similar microcode, then the scope of protection must be narrow. It is only that which is different about the expression contained in the Intel Microcode that copyright protects, if anything.

\textsuperscript{186} Virtually all the similarities that Intel had claimed "proved" copying by NEC are also in the Clean Room Microcode, which could not have been copied.
ing not only the Microcode itself, but foreclosing similar computer architectures. This would necessarily mean that no one could make a compatible machine, be it a microprocessor, personal computer or mainframe computer, where the first machine's control function was performed by copyrighted microcode. The copyright owner would then have patent-like protection over the entire machine, hardware and all, without the safeguards provided under the patent law.187

IV. CONCLUSION

This law suit was begun by NEC to determine, judicially, whether Intel should be permitted to deny to the market superior products competitive with the 8086 and 8088 by an extension of the scope of copyright protection when Intel has consciously decided not to upgrade these two products to meet the needs of the market. Monopoly rights were sought by Intel for the control portion of the 8086/8088 microprocessor in order to keep a better product from the market. If Intel's position is sustained it is unclear who the winner will be. Just as many important copyrights can be claimed by foreign companies as by United States companies. But it is clear who the loser will be: United States manufacturers of products using microprocessors who will be forced to pay whatever price companies free from competition care to charge for inferior products; and United States consumers to whom those prices will be passed and who will be denied state-of-the-art products.

187. See supra notes 34 to 35 and accompanying text.
## Microcode Formats

### 8086/8088 Format

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>Type</th>
<th>a</th>
<th>b</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Condition:** 0-15
- **F**

### V20/V30 Format

<table>
<thead>
<tr>
<th>SOURCE 1</th>
<th>DEST. 1</th>
<th>SOURCE 2</th>
<th>D2</th>
<th>F</th>
<th>W</th>
<th>E</th>
<th>TYPE</th>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **ALU Operation**
- **Tmp**
- **R**

- **Condition:** 0-15
- **INT. CTL**
- **EXT. CTL**
- **SR**