The Case Against Copyright Protection for Programmable Logic Devices

Stephen C. Bishop

Follow this and additional works at: https://digitalcommons.law.uw.edu/wlr
Part of the Computer Law Commons, and the Intellectual Property Law Commons

Recommended Citation
Available at: https://digitalcommons.law.uw.edu/wlr/vol68/iss1/4

This Comments is brought to you for free and open access by the Law Reviews and Journals at UW Law Digital Commons. It has been accepted for inclusion in Washington Law Review by an authorized editor of UW Law Digital Commons. For more information, please contact cnyberg@uw.edu.
THE CASE AGAINST COPYRIGHT PROTECTION FOR PROGRAMMABLE LOGIC DEVICES

Stephen C. Bishop

Abstract: Several commentators have argued that copyright protection should extend to protect logic equations incorporated in a type of semiconductor chip called a programmable logic device (PLD). They reach this result by analogizing to the storage of computer software in memory chips, an embodiment that is currently protected under the copyright laws. This Comment analyzes logic equations incorporated in a PLD with respect to the copyright statute, utilitarian device doctrine, and the legislative history of the Semiconductor Chip Protection Act. It concludes that copyright protection should not extend to protect the logic equations incorporated in a PLD.

Copyright law and computer technology have always had an uneasy marriage. Almost three decades ago, a need developed to provide some form of intellectual property protection for one of the rapidly developing areas of computer technology—computer software.1 Congress eventually opted to adopt the traditional mechanism of copyright law rather than create a sui generis form of protection.2 This shotgun ceremony left the courts with the task of applying centuries-old copyright law to the varied forms and expressions of computer software. What has resulted from this marriage of old doctrine to new technology is an often difficult distinction that courts must make between the hardware and software components in a computer. Copyright protection currently extends only to computer software, and not to hardware.3

Now, however, even the tradition prohibiting copyright protection from extending to computer hardware is being challenged. The vehicle of this challenge is the programmable logic device (PLD), a type of computer chip used in digital microprocessor systems. As a computer chip, the PLD is itself a hardware device. However, the logic equations which are used to program a blank PLD do not easily fit within the traditional definition of software. If logic equations are considered analogous to software, when incorporated into the chip they would be

---

1. The genesis of these problems was the Copyright Office's decision to allow registration of a computer program under the "rule of doubt" in 1964. National Commission on New Technological Uses of Copyrighted Works, Final Report 82 (1979) [hereinafter CONTU Report].
2. See infra note 29 and accompanying text.
3. The Copyright Office rejects registrations for the design of semiconductor chips as well as the chips themselves. See infra notes 48–50 and accompanying text.
protected under the copyright laws. Alternatively, if logic equations are viewed as a language that merely describes the hardware configuration of the chip, then copyright protection should not extend to the programmed PLD. Because logic equations do not clearly fit into either category, the legal community must once again grapple with determining whether the programmed device is protectable under the copyright laws, and if so, to what extent.

This Comment argues that copyright law should not encompass logic equations as incorporated in the PLD. Initially, the Comment analyzes the incorporation of logic equations in the PLD under the literal words of the copyright statute. Next, it examines logic equations as incorporated in the PLD with respect to the axiom that copyright protection should not extend to inseparable elements of utilitarian articles. Due to the construction of the PLD, the expressive elements of logic equations are not separable from their function once incorporated in the chip. Additionally, the legislative history of the Semiconductor Chip Protection Act demonstrates that Congress and the Copyright Office have set a precedent incompatible with the extension of copyright protection to descriptive logic equations. This Comment, therefore, concludes that logic equations as incorporated in the PLD should not be protected under the copyright law. Instead they properly are, and should continue to be, protected under the patent laws.

4. For example, a computer memory chip is a hardware device which can store a computer program; the information incorporated in the chip is protected under the copyright laws. See infra notes 32-44 and accompanying text.

5. See Mark F. Radcliffe, The Future of Computer Law: Ten Challenges for the Next Decade, COMPUTER LAW., Aug. 1991, at 1, 8. The courts may eventually have to resolve whether copyright protection should extend to logic equations incorporated in the PLD, as at least two lawsuits have been filed that claim infringement of copyrights on the "programs" contained in PLDs. The first was Alloy Computer Products v. Ultratek Corp., No. 87-06993 (C.D. Cal. filed Oct. 20, 1987), in which a manufacturer and distributor of a computer board filed a complaint for copyright infringement against a competitor who manufactured and distributed a similar board. The complaint alleged copying of the "computer programs" contained in a type of PLD chip. See Daniel R. Siegel & Ronald S. Laurie, Beyond Microcode: Alloy v. Ultratek—The First Attempt to Extend Copyright Protection to Computer Hardware, COMPUTER LAW., Apr. 1989, at 1. The second was Intel Corp. v. Advanced Micro Devices, Inc., Civ. No. A-91-CA-800 (W.D. Tex. filed Oct. 9, 1991). Intel claimed that a competitor had infringed its copyright by using a control program stored in a programmable logic array on a microprocessor. The case was transferred to the Northern District Court of California. See Michael Slater & Rich Belgard, Intel Claims AM386 Infringes PLA Copyright, MICROPROCESSOR REP., Oct. 30, 1991, at 11; Intel Sues AMD for Infringement of Microcode and "Control" Program; Some AMD Antitrust Allegations Against Intel Dismissed as Time-barred, COMPUTER LAW., Feb. 1992, at 36.
I. THE FOUNDATIONS OF COPYRIGHT PROTECTION

As an aid in understanding how logic equations incorporated in the PLD might fit into the copyright scheme, it is helpful to examine how copyright law currently handles computer software. This section provides an overview of how computer software is written, translated, and used by a computer. It introduces the various forms that computer software can take, and the media in which it can be stored. The section then examines the initial statutory grant and various court interpretations that extended copyright protection to computer software. Finally, the section provides a technical look at a PLD and its operation to understand the conceptual difficulties involved in distinguishing logic equations as incorporated in the PLD from computer software contained in a memory chip.

A. A Computer Software Primer

Modern computers are split into two parts, the hardware and the software. The hardware is the collection of physical components that make up a computer system. At the heart of the hardware in most personal computers is a microprocessor, a semiconductor chip that manipulates all of the data within the computer. By itself, however, the microprocessor can do nothing. It is the software, or computer program, which instructs the microprocessor how to manipulate data to bring about a desired result.

Computer software may be expressed in any of three different levels of computer language: source code, object code, or microcode. High-level languages are the easiest to use, and consist of various combinations of words and symbols which resemble English. While each statement expressed in a high-level language appears to be a single command, it will typically require the computer to perform several smaller steps to reach a desired result. Mid-level languages, called assembly languages, use alphanumeric names to allow a programmer

7. A microprocessor is defined as "the controlling unit of a microcomputer, laid out on a tiny silicon chip and containing the logical elements for handling data, performing calculations, carrying out stored instructions, etc." KENNETH L. SHORT, MICROPROCESSORS AND PROGRAMMED LOGIC 2 (2d ed. 1987).
8. VERGARI & SHUE, supra note 6, at 7. Congress adopted a slightly different definition for computer software in the Copyright Act. See infra note 31 and accompanying text.
9. Examples of high-level computer languages are C, BASIC, PASCAL, and FORTRAN. ELLIOT B. KOFFMAN, PROBLEM SOLVING AND STRUCTURED PROGRAMMING IN PASCAL 7–8 (2d ed. 1985).
to instruct the computer to perform some of these smaller steps.\textsuperscript{10} Most computer programmers will write programs in a high-level programming language or assembly language; in this form programs are referred to as source code.

For a computer to use the source code, a separate program called a compiler must translate it into object code. The resulting object code is in a machine language of binary "1s" and "0s",\textsuperscript{11} and is the only form that the computer can directly manipulate. Object code may be stored on some form of permanent storage device,\textsuperscript{12} from which it can be loaded into a memory chip contained in the computer.\textsuperscript{13}

Finally, when the object code is loaded into a computer, it instructs the microprocessor to perform operations that result in the input, output, and processing of data.\textsuperscript{14} In performing these operations, even the object code does not provide enough information to the microprocessor. For each object code instruction, special microcode instructs the microprocessor to perform several smaller steps.\textsuperscript{15} Microcode controls data manipulation within the computer at the very lowest level. Thus, it is contained in a small memory area in the microprocessor.

B. An Overview of Copyright Protection for Computer Programs in Various Forms and Media

1. General Copyright Doctrine

The copyright statute sets forth a simple rule for copyrightability: a copyrightable work is an original work of authorship fixed in a tangi-


\textsuperscript{11} Computer operation is based on the binary number system which involves only two electrical states, usually represented as "1" and "0". These states are manipulated by the computer using the rules of Boolean algebra, first described in 1854 by George Boole. M. Morris Mano, Digital Design 36 (1984).

\textsuperscript{12} Some examples of permanent storage devices are floppy disks, magnetic tapes, and optical disks.

\textsuperscript{13} This form of storage is located inside the computer. A memory chip can come in many different forms including a RAM (Random Access Memory), ROM (Read-Only Memory), or EPROM (Erasable Programmable Read-Only Memory). Short, supra note 7, at 20.

\textsuperscript{14} For example, a word processing program expressed in object code and contained in a computer's memory tells the computer's microprocessor how to manipulate data so that documents can be created and edited on the computer screen.

ble medium of expression.\textsuperscript{16} The three key elements to this definition are originality, work of authorship, and fixation in a tangible medium.

Generally, the originality requirement is not very stringent. The work must only “owe[] its origin to the author, i.e., is independently created, and not copied from other works.”\textsuperscript{17} The requirement that the subject matter must be a work of authorship is also not very restrictive. The Act specifically lists eight broad categories of works of authorship.\textsuperscript{18} However, the list is only illustrative, and does not necessarily limit the scope of works that the Act was meant to protect.\textsuperscript{19}

The fixation requirement presents the toughest hurdle to satisfy, especially when dealing with computer software. For copyright protection to extend to a work, the work must be fixed in a tangible medium of expression. Fixation occurs when a work’s “embodiment in a copy or phonorecord, by or under the authority of the author, is sufficiently permanent or stable to permit it to be perceived, reproduced, or otherwise communicated for a period of more than transitory duration.”\textsuperscript{20} Further, the perception, reproduction or communication of the work may be performed directly or with the aid of a machine or device.\textsuperscript{21}

The Copyright Act contains two statutory mechanisms, the idea/expression dichotomy\textsuperscript{22} and the utilitarian article doctrine,\textsuperscript{23} that limit the elements that are protected in a work. The idea/expression dichotomy states that it is an "axiom that copyright protection extends only to particular 'expression,' not to concepts or 'ideas' that may be abstracted from concrete embodiment."\textsuperscript{24} Patents, not copy-

\textsuperscript{16} 17 U.S.C.A. § 102(a) (West 1977 & Supp. 1992). The full text of the statute states: “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.”Id.

\textsuperscript{17} 1 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 2.01[A] (1992).


\textsuperscript{21} Id. § 102(a).

\textsuperscript{22} The idea/expression dichotomy was first enunciated by the Supreme Court in Baker v. Selden, 101 U.S. 99, 103 (1879), and later codified in 17 U.S.C.A. § 102(b). But see NIMMER & NIMMER, supra note 17, § 2.18[B] n.15 (noting that nothing in the Copyright Act expressly alludes to the full Baker doctrine).

\textsuperscript{23} The utilitarian article doctrine was first adopted by the Supreme Court in Mazer v. Stein, 347 U.S. 201 (1954), and later codified in 17 U.S.C.A. § 101. See NIMMER & NIMMER, supra note 17, § 2.08[B][3].

\textsuperscript{24} MIDWAY MFG. CO. V. STROHON, 564 F. SUPP. 741, 746 (N.D. ILL. 1983).
rights, protect underlying ideas. The utilitarian article doctrine states that even though a pictorial, graphic, or sculptural work which portrays a useful article might be copyrighted, the copyright protection in the work does not extend to an actual useful article created from the work. Instead, copyright protection only extends to the expressive elements that are physically or conceptually separable from the utilitarian aspects of the article.

2. Statutory Sections Pertinent to Computer Programs

The 1976 revision of the Copyright Act made no mention of computer programs. Based on the final report of the National Commission on New Technological Uses of Copyrighted Works (CONTU), Congress rectified this omission by adopting two amendments to the Copyright Act in 1980. These amendments confirmed that copyright applies to computer programs as literary works. Moreover, the amendments also defined a computer program as "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." Although this definition clarified that computer programs were protectable subject matter, it provided little guidance on which types of programs satisfy this definition.

26. 17 U.S.C.A. § 113 (West 1977 & Supp. 1992). The copyright statute defines a useful article as "having an intrinsic utilitarian function that is not merely to portray the appearance of the article or to convey information. An article that is normally a part of a useful article is considered a 'useful article.'" Id. § 101.
27. HOUSE REPORT ON COPYRIGHT, supra note 19, at 55. The Second Circuit has adopted the following test to determine if an element is conceptually separable: "[I]f design elements reflect a merger of aesthetic and functional considerations, the artistic aspects of a work cannot be said to be conceptually separable from the utilitarian elements. Conversely, where design elements can be identified as reflecting the designer's artistic judgment exercised independently of functional influences, conceptual separability exists." Brandir Int'l, Inc. v. Cascade Pac. Lumber Co., 834 F.2d 1142, 1145 (2d Cir. 1987).
28. CONTU REPORT, supra note 1.
30. The report recommended that the copyright law be amended "to make it explicit that computer programs, to the extent that they embody an author's original creation, are proper subject matter of copyright." CONTU REPORT, supra note 1, at 1. Copyright also extends to computer programs as audio-visual works, but such protection is beyond the scope of this Comment. See, e.g., Williams Elecs., Inc. v. Artic Int'l, Inc., 685 F.2d 870, 872 (3d Cir. 1982).
3. Judicial Expansion of Copyright Protection for Computer Programs

Faced with a lack of specific guidance and an often shaky understanding of the technology, the courts have focused almost exclusively on interpreting the literal requirements of the Copyright Act and the wording of the 1980 amendments when determining the scope of copyrightable subject matter. The result has been a slow expansion of copyright protection to the various forms of computer programs and storage media.

A printed source code listing of a computer program has been copyrightable as a literary text ever since the Copyright Office first registered a computer program in 1964. Copyright protection also extends to printed computer program listings in their translated object code versions. The initial question regarding the extent of copyright protection arose when the courts had to determine whether a program loaded into a memory chip was protectable.

In Williams Electronics, Inc. v. Artic International, Inc., the Third Circuit addressed the issue of whether a source code program translated to object code and loaded into a memory device was copyrightable. The court noted that Congress opted for an “expansive interpretation of the terms ‘fixation’ and ‘copy’ which encompass technological advances such as those represented by the [ROM] in this case.” It also noted that it would be inconsistent to create a “loop-hole” that would allow a potential infringer to duplicate the object code as fixed in a computer chip, yet not allow the infringer to copy the program in a printed form because it was copyrighted. The court therefore found the copyright valid for the object code as fixed in the ROM chip.

---

32. See Midway Mfg. Co. v. Strohon, 564 F. Supp. 741, 750 (N.D. Ill. 1983); see also NIMMER & NIMMER, supra note 17, § 2.04[C].
33. Midway, 564 F. Supp. at 750; accord Williams, 685 F.2d at 875–77.
34. See supra note 13 and accompanying text for a description of memory chips.
35. 685 F.2d 870 (3d Cir. 1982).
36. Id. at 877. The CONTU report stated that copyright would protect programs incorporated on a magnetized tape because of a “one-to-one correspondence” between the written program on paper and the program on the tape. CONTU REPORT, supra note 1, at 22. When a computer program is loaded into a ROM, it is possible to access each memory location to determine its contents. Thus, the entire object code version of the program stored in the chip can eventually be read from the device. The Williams court used this technique when determining that the object code stored in ROM was a copy of the written code, thereby satisfying the fixation definition. Williams, 685 F.2d at 876 n.6.
37. Williams, 685 F.2d at 877.
38. Id. This decision was reaffirmed and clarified in Midway Manufacturing Co. v. Strohon, 564 F. Supp. 741, 749–52 (N.D. Ill. 1983). The court examined the storage of object code in a
Recently, copyright law again expanded to protect another form of software. In *NEC Corp. v. Intel Corp.*, the court was forced to determine where microcode fit into the hardware/software spectrum. At trial, the court first examined microcode stored in a microprocessor in light of the statutory requirements of subject matter and fixation. It defined microcode as a series of instructions used to direct the microprocessor, which easily fit within the statutory definition of computer program. The court also implied that the fixation of microcode was similar to the fixation of object code in a ROM, and found that it was "undisputed" that the microcode was fixed in a tangible medium. Finally, the court found unpersuasive NEC's argument that the microcode was a defining element of the computer itself, and thus fell within the utilitarian device prohibition. Consequently, the court found that Intel's microcodes as contained in the microprocessor were protected under the copyright laws.

4. *A Sui Generis Form of Protection for Computer Masks*

An additional legal development pertaining to the grey area between computer hardware and software occurred in 1984. In the late 1970s and early 1980s, Congress held hearings to determine how to protect...
mask works of computer chips. A mask work is a graphic representation of the internal configuration of the circuits in a computer chip. It is used as a stencil in the manufacturing process to transfer a layout of the chip design to the computer chip itself. While the Copyright Office affirmed that the original drawing of the chip layout on a piece of paper was copyrightable, it had reservations about extending the protection to the chip itself.

The Copyright Office testified against extending copyright protection to mask works by citing its continued refusal to register copyrights in both the design or imprinted patterns in semiconductor chips, as well as the chips themselves. The refusal was based on the conclusion that the design of a chip as “formed in semiconductor material is arguably an intrinsically useful part of a useful article.” It felt that an extension of copyright protection to the topology of semiconductor chips would “grant protection to useful aspects of useful articles, which apparently have no separable artistic features.”

The Copyright Office and Congress considered many factors before concluding that a computer chip is solely utilitarian. One factor was the concern that chip layouts are dictated by the function to be performed by the chip, and not by a creative choice from a number of possibilities. A closely related consideration was the increasing use of computer-aided tools to perform the chip layout, thus reducing the creative input of the designer. Congress also looked at the main purpose of mask works. Even if mask works convey information, their primary purpose is in the manufacture of a useful article—a semiconductor chip. Most importantly, the Copyright Office distinguished general semiconductor chips from other memory chips based on the


47. 1979 Hearings, supra note 45, at 19 (statement of Jon Baumgarten, General Counsel of the U.S. Copyright Office); see 1983 Hearings, supra note 45, at 86 (statement of Dorothy Schrader, Associate Register of Copyrights for Legal Affairs, Copyright Office).


49. Id. at 88.

50. Id. at 92.

51. 1979 Hearings, supra note 45, at 14.


53. HOUSE REPORT ON CHIPS, supra note 46, at 10.
function that they carry out in the computer system. While memory chips only store data, other semiconductor chips have the capability of computing or processing information.\(^5\)

Based on these and other factors, the Copyright Office concluded that both mask works and chips produced from mask works are utilitarian. Because of the policy reasons underlying the utilitarian device doctrine,\(^5\) it argued against the extension of copyright protection to chip designs.\(^5\) Congress responded to this lobbying by enacting the Semiconductor Chip Protection Act of 1984.\(^5\) The only recent \textit{sui generis} method of intellectual property protection, the Chip Act protects the "mask work fixed in a semiconductor chip product."\(^5\)

II. DESCRIPTION OF THE PLD

Courts have extended copyright protection to computer programs contained in memory devices.\(^5\) Commentators have used this extension to argue that because of the similarities between PLDs and memory devices, copyright law should also protect logic equations incorporated in a PLD. It is therefore important to understand the internal configuration, operation, and uses of a PLD to distinguish it from a memory device.\(^5\)

A. Purpose of the PLD

A complex digital system requires supporting hardware to allow the microprocessor to input, output and store information.\(^5\) To link the microprocessor with this hardware requires discrete logic devices or chips.\(^5\) As the system increases in complexity, integration of the various components requires additional logic devices.

---

54. See \textit{1983 Hearings}, supra note 45, at 101. "Other chips have as their primary function to be a computer itself; to process and manipulate information." \textit{Id.}

55. The Copyright Office noted that there is probably no constitutional basis for denying copyright protection to utilitarian works. Rather, it was a matter of policy grounded in the fundamental principle that our copyright law protects only expression. \textit{Id.} at 78-79; see \textit{House Report on Chips}, \textit{supra} note 46, at 8-9.


59. See \textit{supra} notes 32-44 and accompanying text.

60. This Comment attempts to describe the typical application of the PLD. For a further description of PLD uses in digital systems, see generally GEOFF BOSTOCK, PROGRAMMABLE LOGIC DEVICES (1988).

61. See \textit{Short}, \textit{supra} note 7, at 3-4.

62. See \textit{id.} at 456. The three basic logical operations that can be performed on the electrical signals "1" and "0" are AND, OR, and NOT. A single electrical circuit (or "gate") can implement any of these functions. For example, an OR gate might perform the logical OR
The need to design more compact digital systems, while reducing costs, maintaining flexibility, and increasing reliability, led to the development of programmable logic devices. PLD refers to a class of computer chips that can be programmed to replace the function of a group of discrete logic chips. That is, instead of implementing a design using a set of individual logic devices, a computer designer can reduce the entire logic design to a single computer chip.

B. Incorporating Logic Equations in the PLD

Internally, a PLD consists of rows of discrete logic gates which can be selectively connected to each other by programming an array of connecting wires. In order to program a PLD, a designer first creates a set of logic equations specifying the desired logical manipulations to be performed on the input signals. The resulting logic equations are highly dependent on the other hardware components contained in the system.

After creating the set of logic equations specifying how the PLD will manipulate the input signals, the designer uses a PLD assembler to translate the equations to a form used to program the chip. Function on two input lines, and provide the result on an output line. Discrete logic devices are chips that typically implement only a few of these operations. For example, a single chip might contain eight AND gates.

Among the different forms of programmable logic that currently exist are the PLA (Programmable Logic Array), PAL (Programmable Array Logic), GAL (Generic Array Logic), FPGA (Field Programmable Gate Arrays), and ERA (Electrically Reconfigurable Array). David Manners, Logical Devices, ELECTRONICS WKLY., Sept. 18, 1991, at 29. These forms all contain an array of AND gates followed by an array of OR gates on a single chip. SHORT, supra note 7, at 456. This construction allows the input signals to be “ANDed” and “ORed” together in various combinations. Initially, when a PLD is purchased, all of the programmable array elements are connected. The user can then program the device by selectively breaking the connections to realize a desired function between the input signals. Id. at 458. In this way, the designer can implement any logic equation. The very first PLDs were simple devices, with small arrays and limited input and output. With each new generation, the devices have become increasingly complex. The newest PLDs have the capability of providing feedback from the output back to the input lines, see id. at 470, and can contain up to 20,000 gates. Jon Gabay, PLD Development Tools Come of Age, COMPUTER DESIGN, Oct. 1991, at 125.

A PLD assembler is another computer program used to translate the logic equations. It is sometimes called a PLD development tool by the industry. Gabay, supra note 64, at 125.

In addition to translating the logic equations to a form understandable to the computer, the PLD assembler manipulates the equations to reduce them to their simplest state. That is, the assembler uses rules of Boolean algebra to minimize and optimize the number of equations to the smallest or most efficient number. Id.
lowing assembly, the simplified equations can be expressed in two equivalent forms: as a simplified set of logic equations which are logically the same as the original equations, or as a fuse map which indicates how to program the connecting array within the PLD to implement the simplified equations.\textsuperscript{68} A fuse map identifies which array elements to disconnect in the matrix connecting the gates within the PLD. Another device then selectively "burns" open the appropriate connections inside the PLD to implement the final logic design. At this point, the logic equations are "incorporated" in the PLD; the PLD will perform the logical manipulations on input signals as described by the equations.

C. Protection for the PLD

As designers' awareness of the PLD's flexibility grows, they increasingly include the chips in system designs.\textsuperscript{69} Developing complex sets of logic equations to encode onto the chip is an involved process which requires expenditure of a significant amount of time and resources. As with any piece of intellectual property that is labor intensive to create, yet is easily copied, companies will desire a legal mechanism to protect the labor incorporated in these works.\textsuperscript{70}

III. THE CASE AGAINST COPYRIGHT PROTECTION

Some commentators have advocated extending the scope of copyright to include protecting logic equations as incorporated in the PLD.\textsuperscript{71} Their argument is largely based on the similarities between "program-like" logic equations that describe the operation and configuration of the programmed PLD, and the lines of code in a computer program that are loaded into a memory chip. Because the copyright laws protect a computer program both in printed form and as fixed in a computer memory chip, these commentators argue that similar protection should apply to the PLD. The first half of the argument, that logic equations should be protected in their printed form, does not

\textsuperscript{68} These forms are functionally equivalent, and are distinguished only for pedagogical purposes.

\textsuperscript{69} It has been estimated that up to 75% of all new systems contain one or more PLDs, Gabay, supra note 64, at 125, and the sale of chips is expected to top $700 million in 1995. Manners, supra note 64, at 29.

\textsuperscript{70} See CONTU REPORT, supra note 1, at 10.

require an extension of current copyright law. It is the second half of the argument, that copyright protection should extend to the equations as incorporated into the PLD, that would entail a radical and unwarranted broadening of copyright law.\footnote{72}

Copyright protection should not extend to logic equations incorporated in a PLD for two reasons. First, the logic equations do not satisfy the requirements of the Copyright Act. While the logic equations incorporated in the PLD appear to be copyrightable subject matter under section 102, such protection would violate the rule that copyright does not extend to utilitarian works with non-separable expressive elements.\footnote{73} Because a programmed PLD is a utilitarian device, it should not be protected under the explicit wording of the Copyright Act.

Second, the logic equations do not fit within the utilitarian device exception for computer object code and microcode. Because logic equations do not directly instruct a microprocessor, they should not be considered a type of computer software. Instead, logic equations should be treated like mask works used to manufacture semiconductor chips. Congress avoided including mask works within the scope of copyrightable subject matter by creating a \textit{sui generis} form of protection.\footnote{74} Because the process used to manufacture a chip from a mask work is directly analogous to using logic equations to program a PLD, Congress should also decline to extend copyright protection to the logic equations as incorporated in the PLD.

Nonetheless, the intellectual labor involved in designing and producing a PLD should be protected. PLDs are used to simplify the design of complex digital systems; these systems have traditionally fallen within the realm of patentable subject matter. Because the PLD merely replaces a set of discrete logic chips, its configuration should likewise be protected so long as it meets the standards of the patent statute.

A. Logic Equations Incorporated in the PLD and the Requirement of Fixation

Of the three key elements of section 102(a) of the Copyright Act, the requirement of fixation is the most difficult to satisfy.\footnote{75} The major-
ity of logic equations will meet the requirements of originality and work of authorship. A designer who independently creates a set of logic equations would almost certainly satisfy the low originality requirement. Printed logic equations would likewise satisfy the rather broad definition of computer program thereby meeting the authorship requirement.

Printed logic equations also meet the fixation requirement of section 102. On paper, the equations are permanent and can be easily perceived, reproduced, or otherwise communicated. The difficult question is whether the logic equations meet the requirement of fixation when incorporated into the PLD. Insight into this problem may be found by looking at how courts apply the fixation requirement to computer software in a memory device.

Courts have held that a computer program stored in a memory chip satisfies the fixation requirement. They have forwarded two related arguments to justify this conclusion: (1) it would create a loophole to allow potential infringers to copy a program from a chip but not from a written version, and (2) there is a one-to-one correspondence between a program and a version stored in the memory.

While this two-pronged analysis works for computer programs stored in memory chips, applying it to determine whether logic equations incorporated in a PLD meet the fixation test presents more difficulty. The reduction of the logic equations to a form usable in the PLD leads to a simplified set of the original logic equations. Just as

---

76. See NEC Corp. v. Intel Corp., 10 U.S.P.Q.2d (BNA) 1177, 1178–79 (N.D. Cal. 1989) (concluding that evidence of an independent effort is sufficient to defeat the argument that there might be only minimal originality in writing microcode because of hardware limitations); supra note 17 and accompanying text.

77. The programmed PLD is directly used by a computer to bring about a certain result. Because the logic equations are used to manufacture the PLD, they can be said to be indirectly aiding the computer in bringing about that result. This would bring the written logic equations within the broad definition of computer program, which only requires that statements be used directly or indirectly in a computer in order to bring about a result. See supra note 31 and accompanying text; see also Siegel & Laurie, supra note 5, at 11 ("[I]t is difficult to dispute the fact that the logic equations, when written on paper, seem to fit within the Copyright Act’s definition of a computer program.") (emphasis omitted). In a similar manner, source code instructions are indirectly operated on by a computer, and fall within the definition of computer program. Midway Mfg. Co. v. Strohon, 564 F. Supp. 741, 750 (N.D. Ill. 1983).

78. In this respect, there is no requirement that a work be intelligible to human beings, as copyright can even protect the written "1s" and "0s" of written object code. See Williams Elecs., Inc. v. Artic Int'l, Inc., 685 F.2d 870, 877 (3d Cir. 1982).

79. See supra notes 34–38 and accompanying text.

80. See supra note 37 and accompanying text.

81. See supra text accompanying note 36.

82. See supra notes 66–68 and accompanying text.
the original logic equations are copyrightable, the written and assembled version of the reduced logic equations are probably copyrightable because they would likewise meet all three requirements of the Copyright Act. Similarly, the fuse map representing the reduced equations should be copyrightable because it is simply another representation of the optimal equations. Once this fuse map is incorporated into the PLD, however, the analogy to computer software becomes strained.

Under the loophole prong used by the courts, the logic equations should be protected because they can be copied once incorporated in the PLD. The courts' paramount concern was the ability of a potential infringer to circumvent the copyright protection of a printed computer program by freely reading the program from the ROM. A similar copying process is not possible with the PLD; a fuse map cannot be directly read from the chip. It is possible, however, to physically remove the top of the PLD and examine each of the connections in order to reconstruct the original fuse map. Using this approach an infringer can arrive at a copy of the reduced logic equations. Under the Williams and Midway courts' loophole test, the ability to freely copy the design would indicate that the logic equations as embodied in the PLD should be protected by copyright.

The logic equations incorporated in the PLD also seem to satisfy the requirement of a one-to-one correspondence between a written form and the copied form. When a computer program is stored in a memory device, the "1" or "0" of the printed object code is stored in an element in the memory chip. In contrast, when the fuse map is used to selectively burn array connections within the PLD, the information in

---

83. This Comment does not discuss the important question of whether the simplified logic equations are actually another form of the original equations, and thus copyrightable. The assembly of logic equations to fuse map cannot be directly analogized to the conversion process of source code to object code. When attempting to reverse assemble, or go from object code back to source code, it is possible to return to an almost identical source code. However, with logic equations, this reverse assembly is not possible. The original logical equations are not determinable because of the many different possible combinations that would generate the optimal form. This can be more easily seen using an example with more familiar arithmetic, rather than logic, equations. Consider a hypothetical assembler that would optimize the arithmetic equation "5 + 5" to the reduced (but equivalent) form "10" for inclusion into a computer chip. If a designer were later to attempt to reverse assemble the reduced form in the chip, it would be impossible to determine whether the initial form was "7 + 3," "8 + 2," or even "15 - 5." Since this Comment focuses on the logic equations as incorporated in the PLD, a full discussion of whether the preliminary step of optimization precludes copyrightability is beyond its scope.

84. In a similar manner, object code is copyrightable, even though it is a version of source code translated to "1s" and "0s" for use by the computer.

85. See Williams Elecs., Inc. v. Artie Int'l, Inc., 685 F.2d 870, 877 (3d Cir. 1982).
the logic equation only implements a connection to a logic gate that performs the desired logical manipulation. Only through the joint interaction of the array connection and the logic gate is the function of the logic equations implemented. Finding a one-to-one correspondence between the written logic equations and the version incorporated in a PLD would, therefore, require concluding that the physical gate connections in the device are an equivalent expression of written equations that specify the logical manipulation performed by the gate.\(^6\) Logic equations as incorporated into the PLD would technically pass this one-to-one correspondence test. It is this conceptual step, however, which results in conflict with the prohibition against copyright protection extending to utilitarian devices.

B. Logic Equations as Incorporated in the PLD Fall Within the Useful Article Prohibition

Although logic equations as incorporated in the PLD arguably meet the statutory requirement of fixation, the programmed PLD should not be protected under the copyright law because it falls within the utilitarian device prohibition.\(^7\) A useful article has been variously defined as an article whose purpose is not merely to convey information,\(^8\) or one in which the designer's judgment is constrained by functional considerations.\(^9\) The programmed PLD operates in a manner that falls within either definition.

The fixation analysis discussed above—that satisfying the statutory definition requires equating a written logic statement with the physical hardware that embodies that statement—leads to the conclusion that the PLD falls within the first useful article definition. When loaded into the PLD, logic equations are incorporated as actual physical elements, in this case as connections between gates. These physical connections, in conjunction with the logic gates, perform logical operations on input signals. By applying a signal to the input lines of the device, the PLD performs the appropriate logical manipulations and outputs the result. The information contained in the output signals, however, derives from the input signals through logical operations described by the logic equations. The purpose of the PLD is to

---

86. Another commentator has described it as a finding that the logic device which performs a function embodies the corresponding instruction to that function. Siegel & Laurie, supra note 5, at 13.
87. See supra notes 22–27 and accompanying text; see also Samuelson, supra note 38, at 732, for an argument of why copyright protection should not extend to utilitarian works.
88. See supra note 26 and accompanying text.
89. See supra note 27 and accompanying text.
Programmable Logic Devices

perform these logical manipulations on the signals to generate output control signals to the computer; it is not designed to provide any information on its own. Because logic equations describe this manipulation, and do not contain information themselves, when incorporated in the PLD they fall within the prohibition of copyright protection for utilitarian devices.

The PLD can also fall within the useful article definition because of the limited artistic choices a designer makes in writing the logic equations.\(^9\) The PLD is designed to replace several other discrete logic devices. When a designer first develops the original logic equations for a PLD, the expressive elements in the design are minimal. Because a designer typically uses a PLD to link together the microprocessor with surrounding peripheral devices, the hardware configuration and desired functionality imposes constraints on the freedom in choosing equations.\(^9\) Since the goal is a working integration of all hardware in the system, a designer has little chance for any creative expression. The lack of substantial separable expressive elements in the resulting logic equations precludes their copyrightability under the utilitarian device doctrine.\(^9\)

\(^9\) A method of creation test is usually associated with the idea/expression dichotomy and the rule that if there are few ways to express a certain idea, the expression merges with the idea and is unprotectable. As such, the courts usually treat the test as a question of infringement, rather than copyrightability. See NEC Corp. v. Intel Corp., 10 U.S.P.Q.2d (BNA) 1177, 1179 (N.D. Cal. 1989). However, it also provides support to a utilitarian argument because it demonstrates that a designer is seeking to design a functional object, rather than an expressive program. See Midway Mfg. Co. v. Stroh, 564 F. Supp. 741, 751 (N.D. Ill. 1983) (considering, but rejecting, a method of creation test that would examine whether the programmer envisioned the "architectural structure of the ROM," or the "flow chart of operations which the program would perform" to determine if object code is protectable in ROM).

\(^9\) Although this approach might be evidence of the utilitarian nature of a PLD, it is not dispositive. The NEC court considered a merger argument with respect to microcode, where the physical construction of the microprocessor largely dictates the resulting code. It concluded that the limitations imposed by hardware did not make the microcode unprotectable, but only limited the scope of copyright protection to identical copying. NEC, 10 U.S.P.Q.2d (BNA) at 1188–89.

\(^9\) Although the PLD satisfies both definitions of a useful article, a challenge to the characterization could be based upon the wording of the section which codified the utilitarian device doctrine. Section 113 of the Copyright Act states that only pictorial, graphic, or sculptural work can be incorporated in a useful article. 17 U.S.C.A. § 113 (West 1977 & Supp. 1992). The section omits literary works, the category in which logic equations would fall in their printed form. See supra note 77 and accompanying text. In response, the definition of useful article in section 101 of the Copyright Act does not have a limitation like section 113 on which works can be incorporated into utilitarian articles. Compare 17 U.S.C.A. § 101 (West 1977) with 17 U.S.C.A. § 113. It defines a useful article as one having a function that is not "merely to portray the appearance of the article or to convey information." See supra note 26 and accompanying text. This definition is broad enough to include computer programs or logic equations. See Samuelson, supra note 38, at 745–47, for several additional arguments that the omission of
C. Logic Equations Are Similar to Mask Works

Commentators have avoided the utilitarian device prohibition by arguing that logic equations are similar to computer software. This analogy fails because the PLD is used differently from a memory chip within a computer. Instead, logic equations are closer to a hardware descriptive language, and should therefore be treated like mask works for the purpose of copyright protection.

1. Logic Equations Do Not Fall Within the Judicially Created Exception for Computer Software in the Utilitarian Works Category

Although expressive, a computer program is also inherently utilitarian because it directs a computer's operation. When analyzing whether copyright law protects object code and microcode, courts have rejected the argument that a computer program stored in a ROM is utilitarian in nature. Instead, they have consistently cited the CONTU majority statement as evidence that Congress considered programs nonutilitarian: "Programs should no more be considered machine parts than videotapes should be considered parts of projectors or phonorecords parts of sound reproduction equipment . . . . That the words of a program are used ultimately in the implementation of a process should in no way affect their copyrightability."

This limited exception for object code and microcode should not extend to the logic equations because once incorporated into hardware their use is fundamentally different. When incorporated into the PLD, logic equations create a hardware device that is instrumental to the proper function of the computer. A computer program is similarly loaded into a memory chip to provide detailed instructions to a computer. The difference between these two cases is that where a computer program relies on external hardware to fulfill its function, the PLD is designed to perform its operations internally.

A computer program in a ROM chip cannot perform the task for which it was written by itself. Instead, the microprocessor must read the instruction from the memory and implement it. The program's

---

literary works from the list does not exempt computer programs from application of the utilitarian device doctrine.

93. See, e.g., 2 PAUL GOLDSTEIN, COPYRIGHT § 8.5 (1989); Samuelson, supra note 38, at 727–49.

94. NEC, 10 U.S.P.Q.2d (BNA) at 1179; Midway, 564 F. Supp. at 751.

95. CONTU REPORT, supra note 1, at 21; see Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1252 (3d Cir. 1983), cert. dismissed, 464 U.S. 1033 (1984); NEC, 10 U.S.P.Q.2d (BNA) at 1179.
Programmable Logic Devices

purpose is only fulfilled after the microprocessor finishes the task expressed in each line of code. In contrast, the operation of the PLD is internal to the device and does not rely on the microprocessor for implementation. The desired task of manipulating logic signals is performed inside of the PLD. Whereas a computer program directs the computer to reach a desired result, the PLD actually reaches a desired result. Consequently, the PLD should be considered part of the machine (the computer hardware) rather than the software.96 Copyright protection does not extend to machine parts, and therefore courts should not extend copyright protection to the logic equations as incorporated in the PLD.

2. Congress Has Impliedly Rejected Copyright Protection for Descriptions of Chip Layouts by Enacting the Semiconductor Chip Protection Act

In place of a comparison with computer software, a direct analogy can be drawn between the process used to design and manufacture the PLD and the method used to create computer chips from mask works. Instead of drawing a mask work, a PLD user creates a functional description of the computer chip using logic equations. The end result is the same whether the final chip is created from a diagrammatic mask work or from a written list of logic equations: a copyrightable work on a piece of paper describes the eventual layout of a utilitarian computer chip.97 Because mask works and logic equations are very similar, the copyright law should treat them identically.

With mask works, Congress rejected the contention that copyright protection should extend to chip layouts by enacting the Semiconductor Chip Protection Act of 1984.98 When testifying at the hearings on the Act, the Copyright Office supported sui generis legislation rather than an extension of copyright protection.99 The Copyright Office considered sui generis protection superior because it was consistent with several fundamental principles of traditional copyright law, all of which revolved around the prohibition against copyright protection for utilitarian devices.

96. See supra notes 6–8 and accompanying text for a comparison of hardware and software definitions.

97. The Semiconductor Chip Protection Act would not protect the logic equations used to create the PLD because the scope of the act is expressly limited to mask works. 17 U.S.C.A. § 902(a)(1) (West Supp. 1992).

98. See supra notes 45–58 and accompanying text.

The Copyright Office’s arguments rejecting copyright protection for chip masks incorporated in semiconductor chips apply equally to logic equations incorporated in PLDs. Like mask works, the surrounding hardware and intended function of the chip severely constrain the creation of logic equations. Logic equations are also being increasingly designed with the aid of computer development tools. Both of these restrictions in the design process drastically limit the amount of creative expression that a designer may incorporate into a design. Under the utilitarian device doctrine, copyright protection would not extend to non-expressive decisions that are dictated by the hardware or the design tools.

Most importantly, like semiconductor chips which embody mask works, the primary function of the PLD is computational. A PLD is designed to replace discrete logic devices, and is used to process electrical signals based upon rules laid down by the logic equations. Because a programmed PLD is not used to merely store information for eventual use by a microprocessor or other device, the law should treat it as a utilitarian semiconductor chip rather than a memory device. It should therefore not be protected by the copyright laws.

D. Logic Equations as Incorporated in the PLD Should Be Protected Under the Patent Laws

The Copyright Office has noted that there is no constitutional problem per se with extending copyright protection to utilitarian articles. The prohibition, however, has been a “fundamental principle of our copyright laws, adhered to for the nearly 200 years of their existence.” The utilitarian device doctrine recognizes a balancing between the scope and purpose of the copyright and patent laws. It represents a societal judgment that non-novel useful articles should be allowed to be copied and used by all.

The ability to copy the logic equations incorporated in a PLD does not warrant upsetting this balance. The function of a PLD, if it meets the heightened standards of the patent statute, could be protected by a

100. See supra note 51 and accompanying text.
101. See supra note 52 and accompanying text.
102. See supra note 54 and accompanying text.
103. See supra notes 61–64 and accompanying text. Discrete logic devices are not copyrightable. See supra notes 48–50 and accompanying text.
104. 1983 Hearings, supra note 45, at 79.
105. HOUSE REPORT ON CHIPS, supra note 46, at 8.
106. Id.
Programmable Logic Devices

Patent protection has always been available for designs involving discrete logic devices. Copyright protection has been consistently denied for these situations. To allow copyright protection to extend to a single device that is simply a replacement for many discrete devices seems anomalous. Instead, all logic designs, whether embodied as a set of discrete logic chips or contained in a single PLD, should be protected under the patent laws.

IV. CONCLUSION

The analogy of the incorporation of logic equations in a PLD to the storage of computer software in a memory device is a false one. The method of creating the equations, their purpose, and the operation of the PLD in the computer are very different from the role of a computer program stored in a ROM. Because of this difference, logic equations as incorporated in a PLD should be considered part of the computer's hardware and fall within the prohibition against copyright protection for utilitarian devices.

The Copyright Office and Congress have foreshadowed this conclusion by denying copyright protection to mask works as embodied in a semiconductor chip. Unlike a computer program in a memory chip, a PLD is designed to process information based upon a set of logical relations described by a designer. Based on the arguments of the Copyright Office, it can be inferred that the copyright law should not protect the logic equations as incorporated in a PLD. Instead, as has been traditionally the case for utilitarian works and for logic designs, the patent laws should protect logic equations as incorporated in the PLD.

107. See, e.g., U.S. Patent No. 5,043,879 (PLA Microcode Controller); U.S. Patent No. 5,012,163 (Method and Apparatus for Gamma Correcting Pixel Value Data in a Computer Graphics System). Both patents incorporate a PLD as a key system element.

108. See supra notes 48-50 and accompanying text.