INHERENTLY TRANSITORY TRANSISTORS: LEGAL, FACTUAL, AND POLICY ISSUES WITH "RAM FIXATION" IN COPYRIGHT

by
Andrew Lloyd

Submitted in partial fulfillment of the requirements of the
King Scholar Program
Michigan State University College of Law
under the direction of
Professor Sean Pager
Spring, 2010

When the United States Congress passed the Copyright Act of 1976, it took the first step toward removing the formalities previously required of creators seeking to protect their work in favor of a new standard, by which "[c]opyright protection subsists . . . in original works of authorship fixed in any tangible medium of expression."[1] It further defined a work as being "fixed" in a tangible medium when its embodiment is "sufficiently permanent or stable to permit it to be perceived, reproduced, or otherwise communicated for a period of more than transitory duration."[2]

In so doing, the Congress kept its new copyright act within the Constitutional grant of authority, to provide rights to "authors" in their "writings,"[3] while still moving toward the international standard under the Berne Convention of abolishing formalities.[4] Its views on the subject were likely informed by a 1973 case in which the Supreme Court stated that "writings" could be "interpreted to include any physical rendering of the fruits of creative intellectual or aesthetic labor."[5] Congress was therefore drafting to reach the extremities of that contemporary precedent, utilizing the generally accepted broad construction of the Constitutional language.[6]

While no Supreme Court case exists so holding, fixation is generally accepted as a requirement with Constitutional dimensions.[7] Potential policy rationales behind the Constitutional requirement include evidentiary issues, third-party notice, an indication of the author's intentions, a means of increasing availability of the work by placing it in a lasting form, and reduction of the scope of copyrighted works.[8] Intuitively, fixation helps to prevent evidentiary problems by ensuring that a work exists in some producible form before the author can ask a court to enforce remedies against an alleged infringer of that protected expression. Its notice function is related to the availability aspect; if a work exists in some physical form, it is available for others to observe and determine whether their own expression is potentially infringing. Similarly, the policy of reduction of scope has a similar rationale to the one related to authorial intention: because works can only by copyrighted by being fixed, fixation provides at least some evidence that the author created the expression due to the motivation of obtaining a copyright (particularly if the work in question is not inherently fixed, such as by videotaping a performance), while the requirement itself limits the scope of copyright and its intrusions upon speech.

By defining fixation in a tangible medium as requiring some "permanence,"[9] Congress removed from copyright things that were never considered as copyrightable (such as conversations or other ephemeral "recordings" such as writings in the sand, in the frost on a window pane, or using ink that will evaporate from the page[10]). The House Report on the bill indicates that the language in the definition of word "fixed" speaking to duration was intended to "exclude from the concept [of fixation] purely evanescent or transient reproductions such as those projected briefly on a screen, shown electronically on a television or other cathode ray tube, or captured momentarily in the 'memory' of a computer."[11]

Sadly, the courts declined to follow the guidance of Congress in interpreting the act. This paper will examine the holdings of the relevant cases, before presenting the facts about the fundamentally ephemeral nature of computer memory[12] (from the conceptual level of Turing Machine tape to the physical reality of modern RAM speeds), and exploring the arbitrary results created by treating it as "fixed," in the process demonstrating that the intemperate decision to press ahead with memory-as-fixation is bad law and bad policy, and should be abandoned.

1. The Case Law

The seminal case holding that loading into computer memory satisfies fixation is *MAI Systems Corp v. Peak Computer*.[13] This case arose from attempts by MAI, a manufacturer of computers and their operating software, to prevent third parties from performing maintenance on their systems.[14] In order to diagnose and service MAI's computers, Peak's employees would turn them on, causing MAI's software to be loaded into memory, so that the technician could view the system's error log and find out what was wrong.[15] MAI argued that Peak's technicians were thus creating "copies" of their software by causing their computers to operate, and because the software license agreements with the end users who had employed Peak's technicians (rather than MAI's) did not allow for third-party use or copying, Peak's actions could be enjoined as a violation of MAI's copyright in the software.[16] Essentially, MAI was attempting to utilize copyright law to restrain competition in the ancillary market for computer servicing.[17] Peak countered by arguing that the loading of software into memory was not a copyright violation because it was not sufficiently "fixed."[18]

The Ninth Circuit agreed with MAI, upholding the district court's grant of summary judgment and holding that loading into RAM was "copying" for copyright purposes.[19] In so holding it found that, because the software was loaded into memory and then the error log was read from memory, "the representation created in the RAM is 'sufficiently permanent to be perceived, reproduced, or otherwise communicated for a period of more than transitory duration,"[20] although it did not address what precisely made the duration "nontransitory" other than its ability to be displayed.[21] Although the court recognized that there was "no case . . . specifically hold[ing] that copying of software into RAM creates a "copy" under the Copyright Act"[22], it nevertheless proceeded without considering the statement of the drafting committee that information "captured momentarily in the 'memory' of a computer" was excluded from its definition of fixation by the transitory duration clause.[23] And, although the court cited some authority which was dubiously supportive at best,[24] it nevertheless felt impelled to declare that "it is generally accepted that the loading of software into a computer constitutes the creation of a copy under the Copyright Act."[25] It concluded by reiterating its statement that "the copy created in the RAM can be "'perceived, reproduced, or otherwise communicated' . . . the loading of software into the RAM creates a copy under the Copyright Act,"[26] once again omitting any reference to the requirement from the same sentence of the statute it was citing requiring "non-transitory duration" to constitute fixation.[27]

On this dubious ground was built the doctrine of RAM fixation. Shortly after the decision in *MAI Systems*, other companies in the business of providing service on MAI computers filed suit for declaratory judgment of non-infringement on the other coast, resulting in the case of *Advanced Computer Services of Michigan, Inc., v. MAI Systems Corp.*[28] As in the first *MAI Systems* case, plaintiffs argued that "the nature of RAM is inherently so ephemeral, so transitory, as to preclude a finding that a "copy" of a program is made when it is transferred from a permanent memory source to the computer's RAM."[29] The judge noted that "although the contents of RAM are, in some respects, ephemeral or transient, it is important to remember that the Act does not require absolute permanence for the creation of a copy."[30]

Unlike the Ninth Circuit, the district court in *Advanced Computer Systems* separated its analysis on fixation between "materiality" and "transitory duration."[31] As regards the materiality of the copy in memory, the court held that the 1980 amendments to the Copyright Act[32] barred the argument that the electrical currents by which information is represented in RAM were not a "material object," because although imperceptible to an observer, the information therein could be perceived with the aid of a machine, that is, the computer itself.[33]

Turning its attention to the requirement of "non-transitory duration," the court first essentially repeated the mistake of *MAI Systems*, concluding that "useful representations of the program's information or intelligence can be displayed . . . .

virtually instantaneously" and that the information stored in RAM was thus "stable" enough under the statute,[34] thus conflating the "non-transitory duration" language with its preceding clause. The court continued, however, that the fact that the representation of the program loaded into memory would be erased upon turning off the power to the computer "confirmed rather than refuted" its position, because:

> one need only imagine a scenario where the computer, with the program loaded into RAM, is left on for extended periods of time, say months or years, or indeed left on for the life of the computer. In this event, the RAM version of the program is surely not ephemeral or transient; it is, instead, essentially permanent and thus plainly sufficiently fixed to constitute a copy under the Act.[35]

Although the court acknowledged that "if a computer is turned off within seconds or fractions of a second of the loading, the resulting RAM representation of the program arguably would be too ephemeral" under the statute, it ultimately concluded that it was unnecessary to determine the boundaries thereof because "MAI's attack focuses on those situations where the computer is left on for a time measured in minutes, if not longer."[36] Concluding that this was sufficient, it cited with approval the analysis of *MAI Systems* and likewise held that the operation of MAI's computers by repair technicians to have violated MAI's copyright by "fixing" a copy of its software in memory.[37]

It was not until 2008, in the case of *Cartoon Network LP, LLP v. CSC Holdings, Inc.*[38] that a court of appeals would address the issue left open in *Advanced Computer Systems* as to what duration was sufficient to constitute a "transitory" presence in memory. Cablevision, a provider of cable TV services, proposed to offer its customers a service it called "Remote Storage DVR."[39] This system would allow customers to record shows in a manner similar to a Digital Video Recorder such as those sold by TiVo, but rather than relying upon a physical piece of hardware shipped to the end user's home, Cablevision's RS-DVR would record the programs on its own servers and stream them to users when they were ready to watch.[40] To do this, the system utilized a buffer which would store a small portion of each incoming program (no more than 1.2 seconds could be stored at any given time), which would then be copied out to disk at the request of any client who had set their preference to record.[41] Plaintiffs, Cartoon Network and a number of other content providers, filed suit challenging this planned system, alleging only direct infringement while the Cablevision waived any fair use defense they might have otherwise asserted.[42] Thus, the issues before the court were limited primarily to those concerning whether or not the planned system was in itself an infringement of copyright.

In the district court, the plaintiffs were successful in arguing that the RS-DVR system violated their copyrights in three ways; most importantly, that the transient storage of the programs in the buffer constituted a "fixation," and thus an infringing copy, of the work being so buffered.[43] In coming to this conclusion, the district court found persuasive the opinion expressed in the *DMCA Report*, a document prepared by the Clinton administration setting policy in copyright, which argued that the only duration requirement was that the embodiment be fixed longer than "so fleetingly that it cannot be copied, perceived or communicated."[44] The Second Circuit, however, read the statute as "impos[ing] two distinct but related requirements: the work must be embodied in a medium . . . (the "embodiment requirement"), and it must remain thus embodied "for a period of more than transitory duration" (the "duration requirement")," and it faulted the district court for having made the same error as the *MAI Systems* court: it "mistakenly limited its analysis primarily to the embodiment requirement."[45] It went on to criticize *MAI Systems* for failing to consider the issue, before finally deciding to construe the holding of that case as being that "loading a program into a computer's RAM *can* result in copying" but not that it "*always* results in copying."[46] In so doing it reimagined the case to have not involved a "dispute that the duration requirement was satisfied,"[47] although the previously discussed language of that case would seem to dispute the point.[48] Addressing the *DMCA Report*, the court correctly pointed out that its interpretation "reads the "transitory duration" language out of the statute,"[49] as the only requirements for fixation would then be those indicated in the clause before it, rendering the "transitory duration" clause wholly redundant.

Finding that there was not a "serious dispute" over whether the copyrighted works were "embodied" in the buffer,[50] the court proceeded to consider whether the temporary storage in the buffer was sufficiently non-transitory to constitute fixation. The court classified the 1.2 seconds that was the maximum any given piece of data would remain in the buffer as "fleeting," and concluded that the copyrighted works were not "embodied in the buffers for a period of more than transitory duration, and are therefore not fixed in the buffers."[51] Thus, Cablevision had not made any "copies" in its buffer under the statute and its RS-DVR system did not constitute a copyright infringement on this ground. Thus, although rightly critical of *MAI Systems*, the Second Circuit ultimately followed it, while setting a lower bound of 1.2 as "transitory" duration loading of data into memory.[52]

The questionable reasoning of the original decisions notwithstanding, they have dominated the jurisprudence of RAM fixation ever since they were handed down.[53] These decisions have generally relied on relatively limited facts concerning the nature and purpose of computer memory: namely that software is "necessarily" loaded into it[54] and that it "must be repeatedly "refreshed" with an electrical charge or [it] will lose the information stored within."[55] While neither of these statements are factually inaccurate, they represent only the very surface of the purpose, nature, and fundamental characteristics of computer memory. It is no surprise, then, that based on a limited understanding of computer functionality and the fundamental purpose of memory, courts have had difficulty coming to grips with the true consequences of declaring RAM storage to constitute fixation.

To better understand the inherent flaws of these decisions, this paper will now present a bottom up view of memory and its purpose, beginning at the most foundational, theoretical level and building up to the actual, modern capabilities of the technology used to embody it. By doing so, one thing will become clear: the fundamental purpose of computer memory is to be ephemeral.

2. The Realities of Computing

We will begin at the most basic level, with "computers" that predated actual computers: the theoretical constructs known as Turing machines that define the theoretical limits of computability and whose basic design the modern general purpose computer emulates.

Some readers are surely questioning this detour into the realm of theoretical mathematics in a paper generally discussing the field of law. So let us first take a brief stopover in the realms of linguistics to discuss the word "grok." In common computer-enthusiast parlance, to "grok" something is to more than simply understand the surface of it; it is to know it fundamentally from the ground up.[56] In order to grok computer function, and the purpose and essential defining characteristics of memory, it is important to have at least a passing familiarity with the theory upon which they are founded. A brief elucidation of Turing machines is necessary to properly explain the factual error in most discussions of computer memory; while a full description of computability theory is far beyond the scope of this discussion, what follows is an attempt to at least present the very rudiments sufficient to understand the purpose and character of memory in a computer system.[57] The goal of this section is to establish that memory is, at its foundational level, a transitory concept.

Let us begin by laying out the basic characteristics of a simple Turing machine. This machine consists of a tape of infinite length, and a tape head that can read and write symbols and move back and forth across the tape.[58] The tape initially contains the "input" to the machine; that is, the data set to be read and manipulated.[59] The machine itself has a decision function that allows it to decide whether to "accept" or "reject" the input string, and it stores data by writing onto the same tape which contains its input.[60]

Variant Turing machines, such as machines utilizing multiple tapes or having a nondeterministic transition functions, can also be designed; these machines, however, are all reducible to the basic machine; thus the set of problems decidable by *any* Turing machine are theoretically decidable by *every* Turing machine variant.[61] Conversely, problems not solvable by one Turing machine are not solvable by any Turing machine.[62]

Certain characteristics of the Turing machine bear emphasis for our purposes. First, the "tape" is used to hold both the initial input to the machine and whatever information it needs to record that is not captured by its internal set of states. This means that the normal operation of the machine will generally result in the information "stored" on the tape being written and overwritten until such time as the machine reaches its final state (be it accept or reject).[63] Second, a Turing machine can execute the functions of other, different Turing machines by modifying its internal state machine. This, in turn, means that, if a sufficiently detailed grammar for inputs is available, a Turing machine could be designed to execute the functionalities of every Turing machine, given the correct inputs. Such a grammar is called a "Turing-complete" language, and the machine capable of executing it is the "universal Turing machine."[64] From this concept, a computing device that can simulate the functions of any other computing device given enough memory and time, the general purpose computer, was born.

The general purpose computer attempts to make the universal Turing machine a reality. Of course, it is not actually possible in reality to build a machine with an unlimited tape, and so our computers are confined to the world of finite "tape," or, as we have been calling it until now, "memory." Computer memory thus shares the important features of Turing machine tape: it is a scratch space that is designed to be rapidly changed and to lack a stable form. In the most common allocation methodology currently in use, when a program is run, it is loaded into memory (onto the tape) in a particular point called the "stack," so called because it is an instance of the eponymous data structure, characterized by a "last in, first out" structure: like a "stack" of books, it grows by placing more data after the data that is already in place, and data is removed in reverse order of addition, like taking the top book off the stack.[65] Inputs to the program that are of an expected size are also stored in the stack, along with any variables the program itself generates to perform its function. Inputs that do not have a predefined size at the time the program is written are loaded into a different part of memory, called the "heap."

Like the tape of the Universal Turing machine, this memory structure lends itself to a methodology wherein sections of it are read and used to operate on themselves or other sections. A few additional consequences of this design bear emphasis. First, because the data in memory includes both the specification of the Turing machine to be run (the "program," in the terms we had already seen), and the input data, it is possible for a program to modify itself while it is running. Second, a running program in memory is actually more than simply a "copy" of the existing program loaded from whatever storage medium it exists on; substantial additional material surrounds it to perform its function. Indeed, many programs have links to outside material that will be loaded and incorporated dynamically at runtime, meaning that the program that is actually copied into memory will look substantially different from the one found on the storage media, or, indeed, an instance of the that program loaded on a machine with different versions of those linked files.[66] Third, the finite size of memory means that it is possible that segments of the program may not be loaded into memory all at once, if the program is larger than the available memory.[67] Finally, it should be clear that the fundamental dichotomy is not, as is sometimes confused by courts and commentators, RAM as opposed to ROM,[68] but rather memory (the contents of the "tape" of the universal Turing machine being emulated by the computer) as opposed to storage (information which is deliberately kept out of RAM and thus in a stable state). The implications of these facts will be explored later.

Memory itself, however, is not monolithic. To wit, modern memory systems are broken into a set of registers, caches, RAM modules, and virtual memory pages, all serving the same purpose. At its smallest, the "registers" of a processor are single-word (that is, single element of the standard data size for manipulation by the processor) memory storage devices.[69] The registers are a fundamental part of the processor, as they are the sources its instructions use to retrieve and store the data it is currently acting upon. All data, then, must flow through the registers, which must be kept limited in number to allow the processor to function at the high speeds common in modern hardware.[70] The next conceptual (although not actual, as we will see) step is external memory, or the familiar RAM modules one generally thinks of as computer memory. Data that needs to be referenced by the computer to operate the program beyond that which can fit in the extremely sparse register space must be stored in memory; because memory is slower than registers, less frequently accessed data is "spilled" from the registers and into memory so that it can be retrieved later.[71]

As mentioned, however, modern computers have additional steps between the registers and RAM. Because the growth in speed of processors has drastically outpaced the growth in speed of DRAM[72], it is no longer feasible to rely only on a single level of memory for slipped registers. Accordingly, modern systems include a series of intermediate caches, usually referred to by level (L1, L2, L3, and in some cases more); these caches are intermediate memory of much faster speed that are used to mitigate the performance cost of reaching data outside the registers.[73] Memory that slips from the registers is then stored in the L1 cache; if it should slip from the L1 cache (by not being accessed by the time enough other memory has cycled through the cache for it to be marked for deletion), it is stored in the L2 cache, and if it slips from there, into the L3, finally ending up back in ordinary RAM memory should it slip from there.[74] Thus a given piece of data could exist at multiple levels between the processor and main memory, depending on how frequently it is being accessed.

A similar concept is used in reverse to dramatically increase the size of memory available to a program, beyond even the bounds of physical memory. The method of "virtual memory" effectively sets aside a portion of the storage device (generally a hard drive—orders of magnitude slower than RAM but also vastly larger) to be used as an extension of memory in the event that a program in memory requires more physical space than is actually available.[75] By organizing sections of disk into "pages" and assigning them to programs, the computer is able to drastically increase the space available for memory, making it possible for programs to manipulate significantly larger data sets than they would otherwise be capable of.[76] Because of certain file system limitations (and the aforementioned differences between what is found in storage and what eventually is used in memory), retrieving information stored in virtual memory is generally faster than retrieving data existing on the disk normally (because it does not need to be processed, only loaded).[77] Virtual memory thus adds an additional layer of caching, to which data is relegated if it is not accessed frequently enough while in regular memory, and is another potential point at which the data might be found to exist.

Virtual memory has advantages even when physical memory is abundant, primarily due to the commonality of multitasking and multithreading in modern applications.[78] A "thread" is the collection of memory and instructions devoted to a program or a portion of a program by the operating system, which is allowed to execute with at least partial independence of other threads.[79] This structure allows the computer to execute multiple programs concurrently, switching between them as they become idle on a single processor computer or, as is more common in modern hardware, executing them in parallel on multiple processors. A thread that is currently idle can be "paged out" to disk and out of physical memory, to be restored when it is ready to be run again. This results in a substantial efficiency gain, as the current state of the thread is preserved (as it would not be in storage) while it does not take additional memory space it would otherwise have to occupy. Thus, virtual memory is a fundamental feature of every modern operating system.

To review, memory is structured in multiple "levels," which, in descending order of speed and ascending order of size, can be divided into virtual memory, physical memory, processor caches, and processor registers. Each will hold some or all of the data and programs currently be operated on or executed by the processor. This structure roughly approximates the "tape" of the universal Turing machine, although, of course, it is finite rather than infinite; its function, however, is the same: to provide inputs to the machine and then serve as the medium upon which the machine will operate.

Let us move further away from the realm of theory and into practice, and examine the actual specifications of real world hardware, using the current generation of Apple's Mac Pro system.[80] The system utilizes a pair of quad-core Intel Xeon processors in its Nehalem line; this line of processors utilizes an individual 32 KB L1 cache and 256 KB L2 cache per core (so 8 in total for this system), along with a shared 8 MB L3 cache.[81] Although the speeds of those caches are not listed, they are certainly orders of magnitude faster than the next level, which is 1066 MHz SDRAM. This means that this (remember, comparatively slow) memory is capable of transferring 8,533 MB per second of data.[82] For storage, it utilizes at minimum a hard drive with a data rate of 3 Gb/s and a 16 MB cache of its own. Converting to matching units, the hard disk has a data rate of about 286 MB per second.[83] These numbers will be used later in the context of comparing the supposed stability of information in RAM to other media that have been considered unfixed.

A final illustration of the behavior of an extremely simply program may prove helpful in pulling all of this information together. Imagine a simple program, designed to take the contents of a one sentence, plain text file, and display them on the screen. For the sake of brevity, a number of aspects of this program will be ignored,[84] in favor of illustrating the general principles through limited examples.

Because displaying things on the screen and opening and reading files are common tasks, they are implemented by calling portions of stock code, integrated into the operating system.[85] Therefore our program as stored on disk will be fairly short; it will simply first call a function from the system to read the contents of the file into its memory section, and then call another function to read the contents of that memory section and display them on the screen. It will look like a very small number of lines of machine code, after it is compiled, referencing those locations for the functions and defining a variable to store the location of the contents of the file in memory.

When the program is launched, it will be loaded into memory, and a "stack" will be started containing its initial set of variables. At this point, however, the operating system will insert into the program the code for the referenced libraries, changing the pointers in memory to point to the newly instantiated functions. When the running program reaches the step whereby it calls the file-loading function, it creates a new entry on the stack, containing the temporary values to be used by that function. That function in turn will request a section of memory, read the contents of the file into it (through the use of a number of temporary variable locations not included in the original program but created through its operation), and then return a pointer to that location to the original function, before deleting its own memory footprint (and that of any subroutines it calls itself) from the stack. The original program replaces a portion of its memory with the new pointer, then calls the output function, and a similar process ensues. The net result is that the contents of memory representing the program bear little resemblance to the scant few lines stored on the disk.

But imagine that the program takes things a step further. Let's say that the program has a portion by which, if it detects certain directives at the beginning of the file, it replaces its display function with one which outputs the file via a printer. The program in memory could do this by replacing its *own code* with a different instruction, by overwriting the section of memory calling the print to screen function with one calling a print to paper one. The result would be the same program from a computer science perspective, but it would no longer contain the same content as our little program file.

These details are somewhat different for data that is loaded into memory rather than programs, but it would not be uncommon for a file in memory to bear very little literal resemblance to the one stored (in compressed form) on disk. Because the case law in this area has primarily been limited to memory "copies" of software, however, that particular context is more directly relevant to the question of their factual basis, and so will remain the primary focus of this paper.

Having now given more context to the actual facts of computer hardware and the reasons behind them, we will next consider the implications that they have for the approaches pursued in RAM fixation and copyright generally.

3. Implications of the Facts of Computer Memory

The first problem implicated by the truth about computer memory is that its very purpose is to be overwritten, erased, and discarded. As demonstrated above, memory, like the tape of a Turing machine, is essentially the computers "scratch space," wherein it toils until it has come up with the result desired by the user, whether that be the answer to a math problem or a completed novella. The statement in *Advanced Computer Services* that "the contents of RAM are, in some respects, ephemeral or transient"[86] is thus proven incorrect; the contents of RAM are, in fundamental purpose and character, ephemeral and transient. Their entire reason for being is to be temporary; to be erased, modified, or discarded at the whim of the program that caused them to be so loaded in the first place.

Nor is the "insight" of the court in that case that a program loaded into RAM and left running "for the life of the computer" would be "essentially permanent"[87] a reasonable answer. In the first place, such an account fails to consider the aforementioned character of software in RAM: that it is constantly in motion, being changed by the process of being run. Such a "copy" fails to meet the requirement of the Copyright Act that it be "permanent or stable"[88] to constitute a fixation; in fact, it has more in common with a derivative work, as the "copy" of the program in RAM has been "transformed" by the "elaborations" and "modifications"[89] made by the computer in the process of executing it. But this leads to a clearly absurd result: if the copy in RAM is a derivative work, by the definition of "work" which specifies that "each version" of a work prepared in different versions "constitutes a separate work,"[90] a person running a program without permission has committed not *one* count of copyright infringement, but potentially *millions of counts per second* that the program is running! This potentially near-infinite liability, combined with the copyright statutory damages for intentional infringement of up to $150,000 per count,[91] would make even a few seconds of running the program generate more damages than the country's Gross Domestic Profits.[92]

Nor is multiple liability for a single action limited to this scenario. As we have seen, memory itself is actually stored in multiple locations, some of which will necessarily contain the same information. Thus, when something on the computer is in memory, it may actually be in RAM, in all 3 caches (and potentially more, if it happens to be in use by multiple processors or cores), some small portion in the registers, and in virtual memory. Should the portion in registers (likely no more than 80 characters of text) happen to constitute the "core" of the work, this would mean that a single RAM "copy" is actually more like six or *more* copies, all potentially actionable under the standards of *MAI Systems*[93] and *American Computer Services*[94] which equate "non-transitory duration" to "able to be read with the assistance of a machine." Such multiple liability for a single action strains credulity.

Beyond that, an additional problem presents itself in that the standard action upon "deletion" within a computer system is not to completely remove the information so flagged, but to simply "forget" where it is.[95] Such "ghost files" could, potentially, be retrieved until they are overwritten. This potentially brings them within the ambit of the definition of "copies," as they can be "perceived . . . with the aid of a machine."[96] In *Cartoon Network*, the system in question actually wrote over its buffer entirely every 1.2 seconds, although it only actively "recorded" any frame for .1 seconds,[97] and so the point was not addressed. Had the frames of video in that case not been so overwritten, but merely deleted within a fraction of a second of entering memory, their "ghosts" would have lingered in memory for some time thereafter. And if they had so lingered for longer than for a "time measured in minutes, if not longer" as were the alleged violations allowed in *Advanced Computer Services*[98] and approved of by the court in *Cartoon Network,*[99] would that have been a "fixation" sufficient to create a "copy" of the works, and thus an infringement? Such an outcome seems ludicrous (imposing liability for infringing material deliberately deleted before it would have otherwise have become "fixed" as a copy), but reading the statute in the same light as those courts would seem to compel it, providing further evidence that attempting to apply the standards of the RAM fixation cases to reality creates nonsensical results.

Nor does their reasoning hold up when applied more broadly. In the article *Writing in Frost on a Window Pane: E-mail and Chatting on RAM and Copyright Fixation*,[100] beyond the eponymous situation, Brandriss lays out three other questionable fixations in the physical world: "disappearing ink, writing in the sand, and writing through the arrangement of Scrabble tiles."[101] Although Brandriss argues that each of them have different characteristics (namely that the ink will disappear of its own accord, the sand will be washed way by the inevitable actions of an outside force, and the tiles will remain until disturbed by another actor[102]), it is difficult to see how *any* of them fail to constitute fixation under the standards laid down by the courts in the RAM fixation cases. In the case of the Scrabble tiles, the tiles could potentially remain undisturbed forever, which, under the reasoning of *Advanced Computer Services*[103] would make it "permanent," and therefore a fixation. Writing in the sand could potentially remain undisturbed for hours if the tide is low (and possible forever should sea levels happen to drop after you inserted the writing; certainly not significantly less likely than a computer being left running a single program without a restart for its entire effective life), far more than the "minutes" the court in that same case found sufficiently non-transitory.[104] And disappearing ink, even on clothing, remains visible for a few seconds[105], which is more than the 1.2 found insufficient in *Cartoon Network;*[106]although this one is at least arguable since it sits within the gray area between the two cases. Still, that it might even be *possible* for such a thing to constitute fixation under the statute indicates a distinct lack of effect being given to the requirement of the Copyright Act that fixations be for more than "transitory duration."[107]

As can be seen from these examples, the standard cannot be properly set by considering how long something *could* remain in place; by that standard, Brandriss' window-pane poetry could be considered "essentially permanent" because it could be prevented from disappearing by being kept at the right temperature. And any standard based on the potential for change would find RAM to be substantially less fixed than most other mediums; for instance, one of the SDRAM DIMMs mentioned earlier can change its contents over a *billion times* per second.[108] It therefore cannot be asserted with a straight face that RAM is somehow *more* stable than these examples of unfixed works in traditional media and should be held to a different, lower standard; if anything, the speed with which the contents of RAM *could* be changed should require a *greater* length of time before they can be considered "permanent or stable" for more than a "transitory duration."[109] One could perhaps argue that writing in the sand or disappearing ink is different because of the "inevitability" of their erasure but this too proves nonsensical; in the first place, disappearing ink does so in reaction to the air and could safely be preserved in a vacuum, making it not "inevitable" at all, and in the second, as noted, the writing in the sand could persist for hours or even longer, depending on tidal patterns.[110]

And in any event, as we have seen from our analysis of the facts and theory of computer memory, it is unstable by nature and should not be considered any more "stable" than those physical examples. An assertion that these things *should* constitute fixation is an assertion that the words "transitory duration" in the statute are meaningless, potentially in contravention of Constitutional precedent.[111]

Thus, it is clear that there are numerous factual and legal flaws in the current doctrine of RAM fixation: it does not accord with a correct understanding of the purpose and character of computer memory, and, when applied to the facts both within and outside of the field of computing, it produces arbitrary and capricious results in contravention of the statute. A correct understanding of the factual background compels the conclusion that information stored in computer memory is not "permanent" or "stable," and thus cannot be considered fixed.

4. Policy Issues With RAM as Fixation

As discussed earlier, fixation is widely considered to be of Constitutional dimensions, and a number of potential policy rationales have been proposed to explain it.[112] Precisely none of these potential policy rationales support the doctrine of RAM fixation.

The first rationale, that fixation serves a valuable evidentiary function by providing tangible proof of the existence of the work at issue, is not satisfied by the ephemeral "copies" made in memory, because, as established, the contents of memory are in constant flux. They therefore cannot be used as "evidence" until they are actually *fixed* in a stable form that will not simply be overwritten through normal operation. Similarly, the contents of memory provide no notice to third parties, because they are not available to be distributed without independent action and fixation in another form which can be transmitted to others.

Nor can placing something in computer memory be said to be an activity indicating that the author values the work and wishes to copyright it. Because memory is a scratch space that is erased both through normal operation in opening and closing files and whenever the machine is turned off, no rational author would ever believe what was written thereon is preserved until the independent action of saving it to a storage medium is taken. And, of course, that lack of preservation without independent action means that information in memory is not preserved for posterity, because in normal operation without storage, it will be lost, and so RAM fixation is not sensible from the perspective of ensuring that cultural objects are preserved, either.

Finally, RAM fixation is especially troublesome if the policy behind fixation is to limit the scope of copyrighted works, because it essentially gives copyright to any work which at any point touches a computer (even if for only in transmission; for instance, a conversation over a Voice-Over-IP phone system, which uses computers to make phone calls), because everything that the computer does flows through its memory. As the scope of digital tools in the field of communication continues to grow, RAM fixation threatens to place a great portion of human communication under the auspices of copyright.

As can be seen, there is no policy argument for fixation which is satisfied by RAM fixation; it is wholly adrift from any theoretical moorings with which one can ground the fixation requirement. However, there are yet other reasons to discard this illadvised notion; we turn next to the consequences it entails in other fields to which it has been applied.

<u>5. Issues Arising From RAM Fixation in Other Contexts</u>

The consequences of the Ninth Circuit's misunderstanding of computer memory and its transitory character are not limited to nonsensical copyright law. Its mistake has consequences in other areas as well. In *Columbia Pictures, Inc. v. Bunnel,*[113] a district court was confronted with the question of whether "the information held in a computer's [RAM] "electronically stored information" under Federal Rule of Civil Procedure 34?"[114] The rule provides in relevant part that a party may seek discovery of "electronically stored information . . . stored in any medium from which information can be obtained either directly or, if necessary, after translation by the responding party into a reasonably usable form."[115] The case at issue involved a request by plaintiff media conglomerates that defendant torrent-site operators turn over server logs containing the IP addresses of the people who accessed their site.[116] A magistrate had granted the motion over objections by the defendant that, because they did not store the logs on disk but only held them in memory, this was not "electronically stored information" under the statute.[117]

The court, in upholding this decision, found that the defendant's argued definition of "stored," which excludes information which could be held in RAM for "as much as six hours" was "unsupported by . . . Ninth Circuit precedent involving RAM."[118] In reaching this decision, the court claimed that "*amici* and Defendants' argument that data in RAM is too ephemeral to satisfy Rule 34's storage requirement is foreclosed by the Ninth Circuit's decision in *MAI Systems Corp.*"[119] It read that case to establish that "RAM is a tangible medium, sufficiently permanent to permit reproduction," and concluded that

"Defendants' argument that RAM holds data for such a short duration that it is not stored subject to later access and retrieval simply has no merit."[120] It also considered such evidence as a dictionary definition of "store" which it interpreted to mean "simply placing the data in the RAM module is sufficient for it to constitute electronically stored information,"[121] and the NCS glossary definition of RAM as a "storage unit."[122]

It failed to consider, however, what this decision actually meant. As was implied by the previous elucidation of computer function, every action a computer takes is controlled by a program in memory. Therefore every action a computer *takes* has some effect on the contents of memory. If the contents of memory are discoverable "documents" of electronically stored information, then the court could potentially impose sanctions under Rule 37(b)[123] for *spoliation* if a party reasonable should have known that it might be sued and failed to retain that information. In other words, essentially every time a company was under circumstances in which it would normally issue a document retention policy in anticipation of litigation, it would have to retain the contents of memory, of every computer, at all times, after it foresaw that possibility. But, as previously mentioned, every action, *including the act of preserving the contents of memory*, would change the contents of memory. Essentially the result of this is that the conclusion that the contents of memory are "electronically stored information" would make it *illegal* to operate a computer while a document retention policy was prudent, because the computer would generate "electronically stored information" faster than that information could be, well, *stored.*

Beyond the obvious absurdity of making the operation of any computer system grounds for discovery sanctions, such a conclusion also has the effect of expanding the scope of electronically stored information above that of physically stored documents, by included inherently ephemeral "storage." In its amicus brief on appeal,[124] the Electronic Frontier Foundation raises this argument, pointing out that, were similar standards applied to physical "documents," they would include such measures as "erasing a whiteboard . . . ask[ing] a litigant to record all telephone calls . . . videotap[ing] all staff meetings, or . . . outfit[ting] every potential witness with a GPS tracking device in order to make a record of their locations at every moment."[125] No court would so hold, it claims, because:

> discovery simply does not reach such ephemera, even if highly relevant and easily collected (snapping photos of whiteboards is simple; many digital telephone systems can readily be configured to record calls; inexpensive video cameras now have massive storage capacities; free programs allow one to track location via GPS features built into many modern cell phones). Yet that type of discovery is precisely what the Server Log Data Order contemplates, solely because the electronic equivalent of such ephemeral communications will necessarily involve the creation of a temporary snapshot that could, in theory, be preserved.[126]

This result, that the equivalent of such onerous restrictions might be placed on a party to litigation, is the result of attempting to apply the reasoning of *MAI Systems* and the other RAM fixation cases, into seemingly analogous fields; if data is "permanent or stable enough to be perceived," it does not, at first blush, seem unreasonable to conclude that it is "stored." But, as we can see, the mistake in the first case becomes more pronounced when its reasoning is applied by analogy.

Notwithstanding the exceptionally unsympathetic defendants,[127] the court's ruling in *Bunnel* extending *MAI Systems* to the discovery context is an indicator of the serious danger of other courts importing its flawed reasoning into other contexts. Although the district court judge downplayed concerns as to the consequences of her decision, noting that "[i]n response to *amici's* concerns over the potentially devastating impact of this decision on the record-keeping obligations of businesses and individuals, the Court notes that this decision does not impose an additional burden on any website operator or party outside this case;"[128] while technically true (as a district court opinion, it is binding only on the parties), the danger of other courts finding its faulty reasoning as persuasive as it found that of the court in *MAI Systems* underscores the danger that such decisions pose simply by remaining on the books. While the inherent impossibility of actually following the standard that *Bunnel*'s decision would impose would presumably make amending the Rules of Civil Procedure to overturn it relatively easy to accomplish (due to pressure from industry on Congress that would result from an interpretation that would essentially ban the PC workstation), it still stands as an indication of the inherent danger of the holding to sound public policy.

One of the fundamental issues raised in *Bunnel*, that nearly every action on a computer results in some transitory storage of data into memory, has broader consequences as well. Because the idea of caching is fundamental to the function of local memory systems, it is no surprise that wide scale networks utilize similar techniques to minimize the amount of time spent waiting for data from other computers (which, as they must not only be read from the disk of that machine onto your own but also traverse the intervening network, is significantly slower than all of the above discussed hardware). In fact, a great deal of the perceived speed of the internet is created by intermediate proxy servers, which cache commonly requested material to reduce the round-trip costs of requests for commonly desired files.[129] Other commonly cached information includes the translation of web addresses to computer addresses ("DNS Caching"),[130] TCP packet caching to prevent loss of data in transmission,[131] and, of course, the well known cache of one's web browser of choice, which stores information from recently visited sites, reducing the amount of time needed to load them on subsequent visits. Naturally, this data all exists for the purely functional purpose of increasing the efficiency of the network as a whole. But it can often found in storage rather than memory due to its infrequent use (as compared to the data in RAM, which is there precisely because it is being acted upon in the immediate term); as a result, it is unquestionably "fixed" under *MAI Systems* and its progeny, and thus potentially any copyrighted data found therein is an infringement.

Congress has provided some safety from this potentially looming liability to service providers, at least, through section 512 of the Copyright act.[132] That section provides in relevant part that cached copies created by an "automatic technical process" are exempt from liability.[133] But this section applies *only* to service providers, not to citizens or operators of networks; while it removes the largest dangers of potential surprise suits due to material that the operator did not even know was stored on his system, it does not wholly resolve the troubles created by cached data. Interestingly, that same section has a provision regarding "Transitory Digital Network Communications,"[134] which provides immunity for copyrighted data transmitted by service providers but not cached, so long as in the course of such "intermediate or transient storage" the material is not maintained for "a longer period than is reasonable necessary for the transmission."[135] This section seems to indicate a different definition of "transitory" than that embraced in *MAI Systems*.

Additionally, as technologies advance, the potential to "design around" RAM fixation doctrine becomes increasingly possible. The RS-DVR system at issue in *Cartoon Network* may well have been one early attempt to do just this; it stored of its own accord only in the earlier discussed buffer, which the court found too "transitory" to constitute fixation.[136] It then created individual copies on a per-user basis upon their request, which the court found to escape liability for copying because "volitional conduct is an important element of direct liability."[137] And finally, it transmitted that "single unique copy produced by that subscriber" to the individual subscriber, it found that the system did not constitute a "performance" of the work under the act, either.[138] Essentially, then, each step of the process of the RS-DVR was designed to evade the law in some specific way; copyright law has become, in the computing context, a flaw to be designed around.[139]

To sum up, the effect of the RAM fixation doctrine on other areas of law also mitigates against it. The danger its possible extension by analogy poses in the discovery context and substantially any other area of law dealing with the "stability" of documents cannot be understated. The rationale underlying it is fundamentally at odds with the basic functionality of the internet with regards to the caching of information to improve speed and reduce traffic. And the current state of the law encourages hardware designers and software programmers to think of copyright as a technical flaw to be designed around rather than an aspect of the rule of law.

6. Conclusion

As we have seen, substantial legal, factual, and policy problems exist in the doctrine of RAM fixation. These problems are so fundamental that I believe they cannot be surmounted; I believe the appropriate conclusion to be drawn is that RAM fixation is a mistaken doctrine that should be overruled.

A more appropriate system for determining whether or not a given work has been "fixed" for copyright purposes should focus less on attempting to calculate probabilities that a given embodiment will remain extant for an arbitrarily chosen amount of time, and more on the reasonable expectations of the artist or author in embodying his or her work in that particular medium. Such a system would reject claims of fixation in "purely evanescent or transient reproductions such as those projected briefly on a screen, shown electronically on a television or other cathode ray tube, or captured momentarily in the 'memory' of a computer,"[140] as the House Report rightly contended to be excluded from the concept of fixation, as well as writings in the sand or frosted glass. To obtain a copyright in such mediums, the artist has a simple recourse: the fixation of an image of them in a photograph. This requirement is not onerous, and would better serve copyright's incentive purpose to "promote the progress of science and useful arts"[141] by requiring artists to keep their work in a form that could actually be observed by subsequent generations. It would also serve the notice purpose, in actually providing stronger evidence of the nature of the copyright allegedly infringed.

This approach rests on the reading of the word "transitory" to include an expectational element; an embodiment is "transitory" under my definition if its owner could not reasonably expect that it will remain so embodied absent additional effort on his part. The "reasonable expectation" standard would be based on the natural properties of the medium in question; if the medium is sufficiently unstable that a rational actor concerned with preservation would record it in a different medium, an author could not reasonably expect it to be preserved without so recording, and therefore its duration is "transitory" until it is so preserved.

This definition is not out of line with our intuitive understanding of "transitory" things; I know, for instance, that although the ice cube on my table during the summer is "transitory," that I can prevent its passing by placing it in the freezer. Further, the results would better accord with those one might reasonably expect: writing on a chalkboard, absent additional circumstances and effort, would not be "fixed;" writing on a paper would be. Writing in a computer program would not be fixed until the "save" command was entered, according with the common safety tactic of saving early and often to prevent data loss (because, of course, until it is saved, the information in your document exists only in the transitory memory of the computer, subject to the capricious whims of Microsoft Word to determine its fate).

Such as system would have clear advantages over the deeply flawed and fundamentally mistaken doctrine of RAM fixation currently dominating the judicial construction of fixation in copyright law. At the very least, the elucidation provided of the factual and theoretical underpinnings of computation should raise serious doubts as to the factual basis on which the cases were decided.

Or, failing that, at least I have hopefully taught you what the word "grok" means.

---

[1] 17 USC § 102

[2] 17 USC § 101

[3] U.S. CONST. ART. I, § 8, CL. 8.

[4] BERNE ART. 5(2) (forbidding the requirement of formalities to obtain copyrights); *see also* BERNE ART. 2(2) (allowing for a fixation requirement for protection)

[5] Goldstein v. Cal., 412 U.S. 546, 561-62 (1973).

[6] *See* 1 NIMMER ON COPYRIGHT 1.08 (discussing the traditional trend of broad interpretation dating back to the first copyright act of 1970).

[7] *See eg.* 1 NIMMER ON COPYRIGHT § 203[B]. Additionally, footnote 23 of that section includes a list of cases citing Nimmer's conclusion that fixation is a constitutional necessity.

[8] *See eg.* Yoav Mazeh, *Modifying Fixation: Why Fixed Works Need to be Archived to Justify the Fixation Requirement*, 8 LOY. L. & TECH. ANN. 109, 118-122 (2009) (setting out policy justifications for fixation); *see also* Stefan Hubanov, *The Multifaceted Nature and Problematic Status of Fixation in U.S. Copyright Law*, 11 INTELL. PROP. L. BULL. 111, 118-120 (2006).

[9] According to 17 USC § 101, the touchstone measures for fixation in a tangible medium are whether the embodiment is "sufficiently *permanent* or *stable*." (emphasis added)

[10] *See generally* Ira L. Brandriss, *Writing in Frost on a Window Pane: E-Mail and Chatting on RAM and Copyright Fixation*, 43 J. COPYRIGHT SOC'Y U.S.A. 237 (1996) (hereafter "Brandriss"), discussing various traditional examples of non-fixation with regard to internet communication. *See also* 1 NIMMER ON COPYRIGHT § 101 at footnote 463 (citing Brandriss's article as examples for potentially non-fixating mediums).

[11] H.R. REP. NO. 94-1476, at 53.

[12] Many courts and commentators use primarily the acronym "RAM" to denote the location of data loaded into a computer for reading or execution. As this paper will attempt to show, "memory" in a computer is the appropriate term of art, which encompasses much more than the Random Access Memory. Thus I will generally use "memory" to refer to what most call "RAM," except when discussing what has come to commonly be called "RAM fixation" in the law.

[13] 991 F.2d 511 (9th Cir. 1993).

[14] Id. at 513.

[15] Id. at 513, 518.

[16] Id. at 517.

[17] Indeed, in the later case of *Advanced Computer Services v. MAI Systems* 845 F.Supp. 356 (E.D.Va. 1994), other businesses servicing MAI computers attempted to bring antitrust claims against MAI for this behavior, although they were ultimately unsuccessful. Id. at 367-70.

[18] MAI Systems, 991 F.2d at 518.

[19] Id.

[20] Id.

[21] An explicit attempt to make "nontransitory" coterminous to the ability to be "reproduced" in the DMCA Report (DMCA SECTION 104 REPORT 111) is discussed and ultimately rejected in *Cartoon Network LP, LLLP v. CSC Holdings, Inc.*, 536 F.3d 121, 127, 129 (2nd Cir. 2008), discussed later in this paper.

[22] MAI Systems, 991 F.2d at 519.

[23] *See* H.R. REP. NO. 94-1476, at 53.

[24] *See* eg Brandriss, 43 J. COPYRIGHT SOC'Y U.S.A. at 250 (pointing out that the case of *Apple Computer, Inc. v. Formula Int'l, Inc.*, 594 F.Supp. 617 (C.D.Cal. 1984) actually appears in context to be indicating that RAM copies are too "temporary").

[25] MAI Systems, 991 F.2d at 519.

[26] Id.

[27] 17 USC § 101

[28] 845 F.Supp. 356 (E.D. Vir. 1994).

[29] Id. at 362

[30] Id. at 362-63.

[31] *See* Id. at 363.

[32] *See* 94 STAT. 3028 (1980) (codified at 17 USC §§ 101, 117).

[33] Advanced Computer Services, 845 F.Supp. at 363.

[34] Id.

[35] Id.

[36] Id.

[37] Id.

[38] 536 F.3d 121 (2nd Cir. 2008)

[39] Id. at 124

[40] Id.

[41] Id. at 124-25.

[42] Id. at 124.

[43] *See* Id. at 125 (discussing the lower court decision).

[44] DMCA SECTION 104 REPORT at 111.

[45] Id. at 127.

[46] Id at 127-28.

[47] Id. at 128.

[48] *See* MAI Systems, 991 F.2d at 518 ("MAI has adequately shown that the representation created in the RAM is "sufficiently permanent or stable to permit it to be perceived, reproduced, or otherwise communicated for a period of more than transitory duration."").

[49] Cartoon Network, 526 F.3d at 129.

[50] Id.

[51] Id. at 130.

[52] Indeed, at id. at 128-29, the court cited *Advanced Computer Systems* with approval, suggesting that one attempting to make some sort of sense of the precedent could claim that the "line" that the *Advanced Computer Systems* was unwilling to draw must lie somewhere between that court's "minutes" (sufficient for fixation) and *Cartoon Network*'s 1.2 seconds (insufficient for fixation).

[53] *See eg.* Quantum Systems Integrators, Inc. v. Sprint Nextel Corp. 338 Fed.Appx. 329, 336-37 (4th Cir. 2009) (affirming *MAI Systems*' status as leading case and noting other district and circuit courts following its ruling).

[54] MAI Systems, 991 F.2d at 157.

[55] Advanced Computer Services, 845 F. Supp. at 362.

[56] *See* The Jargon File, http://www.catb.org/jargon/html/G/grok.html (last visited May 9, 2010)

[57] Those interested in learning more on the subject may find it in MICHAEL SIPSER, INTRODUCTION TO THE THEORY OF COMPUTATION, PWS Publishing Company (1997) (hereafter "SIPSER")

[58] SIPSER at 125.

[59] Id. 126.

[60] Id.

[61] *See* id. at 136-42.

[62] *See generally* id. 151-68 for a discussion of decidability and a discussion of why the "halting problem"—whether or not, with a given set of inputs, a Turing machine will be able to reach an "accept" or "reject" state—is undecidible.

[63] *See* id. at 126-27 for an example Turing machine and its potential outputs onto the tape during its run for an explication of why and how this would occur.

[64] *See* id. at 160.

[65] For a representation of this process in the particular context of a C program compiled and assembled in the MIPS instruction language, *see* DAVID A. PATTERSON AND JOHN L. HENNESSY, COMPUTER ORGANIZATION & DESIGN: THE HARDWARE/SOFTWARE INTERFACE, Second Edition, pp. 156-63, Morgan Kaufman Publishers, Inc. (1998) (Hereafter "PATTERSON").

[66] For a brief overview of the functionality of a linker, *see* Id. at 158-59.

[67] *See eg.* ANDREW S. TANENBAUM, MODERN OPERATING SYSTEMS, Second Edition p. 202, Prentice-Hall, Inc. (2001) (hereafter, "TANENBAUM"), discussing the now-obsolete method of separating programs into "overlays," which would be run in sequence. This procedure is no longer necessary in most computers, because the memory size available outstrips the unloaded size of most programs, obviating its necessity.

[68] *See eg.* Advanced Computer Services, 845 F.Supp. at 362 (contrasting RAM against ROM in determining stability); Douglas J. Masson, *Fixation on Fixation: Why Imposing Old Copyright Law On New Technologies Will Not Work*, 71 IND. L. J. 1049, at footnote 63 (1996) (contrasting RAM against ROM in elucidating its function).

[69] *See* PATTERSON at 109-10.

[70] Id.

[71] *See* id. at 115.

[72] *See* id. at 618-19. The growth rate disparity is so high, in fact, that to display them on the same graph the editors had to make the y-axis (improvement factor) scale logarithmically.

[73] *See* id. at 618-22.

[74] *See* TANENBAUM, pp. 23-24.

[75] *See* id. at p. 202.

[76] *See* id. at p. 202-05.

[77] Id.

[78] *See* id. at 190-202.

[79] *See* id. at 81.

[80] Technical specifications may be found at http://support.apple.com/kb/SP506 (last visited May 9, 2010)

[81] *See* http://www.intel.com/products/processor/corei7/specifications.htm (last visited May 9, 2010).

[82] The SDRAM utilizes a 64-bit bus; because a byte is 8 bits, its peak transfer rate in MB is thus 8 times its data rate.

[83] As the speed is measured in Gb/sec, getting it into the same units requires division by 8 (as before, 8 bits to a byte).

[84] For instance, such a program would tend to function by displaying information to the "console," which in a modern computer system is itself displayed through the window manager functions implemented by the operating system.

[85] In actual practice, it is far more complicated than this; the program would call a function from a "library," which would in turn trigger a file action by the operating system, which would in turn trigger a call to the hardware drive of the storage medium. For simplicity's sake, the example omits these steps, but it should be noted that the result of this is that the program looks even more different than is herein described when in memory.

[86] Advanced Computer Services, 845 F.Supp. at 362.

[87] Id. at 363.

[88] 17 USC § 101.

[89] Id.

[90] Id.

[91] 17 USC 504(c).

[92] Some objection may be raised as to the potential liability under the rubric of volition; a full discussion of that rather murky area of copyright law, however, is beyond the scope of this paper.

[93] MAI Systems*,* 991 F.2d at 519 ("since we find that the copy created in the RAM can be [perceived] we hold that the loading of software into RAM creates a copy").

[94] Advanced Computer Services, 845 F.Supp. at 363 ("Once a software program is loaded into to a computer's RAM, useful representations of the program's information or intelligence can be displayed . . . . virtually instantaneously. Given this, it is apparent that a software program residing in RAM is [sufficient to satisfy the language of the statute including the non-transitory clause]").

[95] *See* TENENBAUM at 428-30.

[96] 17 USC § 101.

[97] Cartoon Network, 536 F.3d at 124-25.

[98] Advanced Computer Services, 845 F.Supp. at 363.

[99] *See* Cartoon Network, 536 F.3d at 128-29 (citing the aforementioned language favorably).

[100] Brandriss, 43 JCPS 237.

[101] Id. at 257-58.

[102] Id.

[103] *See* Advanced Computer Services, 363 F.Supp. at 363.

[104] *See* Id.

[105] *See* http://chemistry.about.com/od/demonstrationsexperiments/ss/disappearink_4.htm (last visited May 9, 2010), discussing the creation and function of disappearing ink.

[106] Cartoon Network, 536 F.3d at 129-30.

[107] 17 USC § 101.

[108] One MT/s is the transfer of one piece of data into or out of memory. Thus, a 1066 Mhz DIMM is able to perform over a billion transfers of memory per second.

[109] *See* the definition of fixed, 17 USC § 101.

[110] An alternative formulation, that these things should not be considered fixed because a reasonable person would not believe them to be, will be taken up at the conclusion of this paper.

[111] *See* the discussion of Goldstien v. Cal., *supra*.

[112] *See supra*, at notes 7 & 8.

[113] 245 F.R.D. 443 (C.D.Cal. 2007).

[114] Id. at 446

[115] FED. R. CIV. P. 34(a)(1)(A).

[116] *See* Bunnel, 245 F.R.D.at 445-46.

[117] Id.

[118] Id. at 446.

[119] Id. at 447.

[120] Id. at 448.

[121] Id. at 446-47.

[122] Id. at 447.

[123] FED. R. CIV. P. 37(b).

[124] Br.for Amici Curiae Electronic Fronteir Foundation, Center for Democracy & Technology, and Public Knowledge in Supp. of Neither Party, Urging Vacatur of the Aug. 24, 2007 Order Denying Defs.' Mot. for Review", *available at* http://www.eff.org/files/filenode/torrentspy/20090203%20AppellantsBrief.pdf (last visited May 9, 2010) (hereafter "EFF Brief")

[125] *See* Id. at 8-9.

[126] Id. at 9.

[127] Id. at 1-2, pointing out that the defendants, in addition to being accused of widespread enablement of copyright infringement, were also eventually sanctioned for "widespread and systematic efforts to destroy evidence."

[128] Bunnel, 245 F.R.D. at 448.

[129] See JAMES F. KUROSE AND KEITH W. ROSS, COMPUTER NETWORKING: A TOP-DOWN APPROACH FEATURING THE INTERNET, Second Edition, pp. 1156-62, Pearson Education, Inc. (2003) (hereafter "KUROSE").

[130] Id. p. 129.

[131] *See* id. pp. 238-48. Although the relevant terminology speaks more of loss-prevention and reliability, one technique employed to ensure it is the saving of packets along intermediate hops to allow for faster retransmission upon loss, which is caching.

[132] *See* 17 USC § 512.

[133] 17 USC § 512(b).

[134] 17 USC § 512(a).

[135] 17 USC § 512(a)(3).

[136] *See* discussion of Cartoon Network in part 1, *supra*.

[137] Cartoon Network, 536 F.3d at 131.

[138] *See* Id. at 139-40.

[139] *See also* Vivian I. Kim, *The Public Performance Right in the Digital Age:* Cartoon Network LP v. CSC Holdings, 24 BERKELEY TECH. L.J. 263 (2009) (arguing that Cablevision skirted the public performance issues through an inefficient, redundant design serving no technical purpose, only their copyright arguments).

[140] H.R. REP. NO. 94-1476, at 53.

[141] U.S. CONST. ART. I, § 8, CL. 8.