

No. 13-298

IN THE
Supreme Court of the United States

ALICE CORPORATION PTY., LTD.,

Petitioner,

v.

CLS BANK INTERNATIONAL AND CLS SERVICES LTD.,

Respondents.

On Writ of Certiorari to the
United States Court of
Appeals for the Federal
Circuit

**BRIEF OF *AMICUS CURIAE* COMPUTER &
COMMUNICATIONS INDUSTRY ASSOCIATION
IN SUPPORT OF RESPONDENTS**

MATTHEW LEVY

Counsel of Record

ALEXANDRA STERNBURG

BRIAN KAHIN

COMPUTER &

COMMUNICATIONS

INDUSTRY ASSOCIATION

900 17th Street, NW,
Suite 1100

Washington, DC 20006

(202) 783-0070

mlevy@ccianet.org

February 27, 2014

TABLE OF CONTENTS

TABLE OF AUTHORITIESiii

INTEREST OF *AMICUS CURIAE* 1

SUMMARY OF THE ARGUMENT 1

ARGUMENT 4

I. As This Court Recognized in *Microsoft v. AT&T*, Software Is Only Meaningful When It Runs on Particular Computer Hardware 4

 A. Patents That Include Software But Are Not Restricted to Specific Computer Hardware Result in Overbroad Functional Claims 6

 B. The Federal Circuit Has Created an Easy Way for Patentees to Evade *Halliburton* and Enforce Functional Claims 7

II. This Court Should Adopt a Test That Requires Software to be Restricted to a Particular Computer Architecture or an Independently Patentable Process or Machine.. 11

 A. If Software in a Patent Claim Is Restricted to a Particular Computer Architecture, It Is Not Abstract..... 12

 B. Means-Plus-Function Form Can Limit a Claim to a Particular Computer Architecture 14

 C. If Software in a Patent Claim Is Not Limited to Particular Computer Hardware, This Court’s Precedent Says That the Software Cannot Be Considered When Assessing Patentability 15

III. The Proposed Test Offers a Clear Solution
That Demonstrates The Invalidity of Alice's
Asserted Patent Claims 18

CONCLUSION 21

TABLE OF AUTHORITIES

Cases

<i>Accenture Global Servs. v. Guidewire Software</i> , 728 F.3d 1336 (Fed. Cir. 2013).....	16, 17
<i>Bancorp Servs. v. Sun Life Assur. Co. of Canada</i> , 687 F.3d 1266 (Fed. Cir. 2012).....	17
<i>Burr v. Duryee</i> , 68 U.S. 531 (1863)	4
<i>DealerTrack, Inc. v. Huber</i> , 674 F.3d 1315 (Fed. Cir. 2012).....	9
<i>Diamond v. Diehr</i> , 450 U.S. 175 (1981)	14
<i>Halliburton Oil Well Cementing Co. v. Walker</i> , 329 U.S. 1 (1946)	<i>passim</i>
<i>In re Alappat</i> , 33 F.3d 1526 (Fed. Cir. 1994).....	12
<i>Mayo Collaborative v. Prometheus Labs.</i> , 132 S. Ct. 1289 (2012)	15
<i>Microsoft Corp. v. AT&T Corp.</i> , 127 S. Ct. 1746 (2007)	2, 4, 17
<i>O'Reilly v. Morse</i> , 56 U.S. 62 (1853)	2
<i>Parker v. Flook</i> , 437 U.S. 586 (1978)	<i>passim</i>
<i>Renhcol Inc. v. Don Best Sports</i> , 548 F. Supp. 2d 356 (E.D. Tex. 2008)	9

<i>York Prods., Inc. v. Central Tractor Farm & Family Ctr.</i> , 99 F.3d 1568 (Fed. Cir. 1996).....	9
Statutes and Other Authorities	
35 U.S.C. § 101.....	<i>passim</i>
35 U.S.C. § 112(f)	<i>passim</i>
Leahy-Smith America Invents Act of 2011.....	14
Patent Act of 1952	8, 11, 14
Bronwyn H. Hall & Dietmar Harhoff, <i>Recent Research on the Economics of Patents</i> , NATIONAL BUREAU OF ECONOMIC RESEARCH, Working Paper 17773 (Jan. 2012).....	3-4
Dennis Crouch, <i>Functional Claim Language in Issued Patents</i> , PATENTLY-O, Jan. 23, 2014.....	10
Mark A. Lemley, <i>Software Patents and the Return of Functional Claiming</i> , 2013 WIS. L. REV. 905 (2013).....	9
Walter J. Boyne, <i>The Wright Brothers: The Other Side of the Coin</i> , WINGS OVER KANSAS, Apr. 20, 2008	7

INTEREST OF *AMICUS CURIAE*¹

The Computer & Communications Industry Association (CCIA) represents large, medium-sized, and small companies in the high technology products and services sectors, including computer hardware and software, electronic commerce, telecommunications and Internet products and services — companies that collectively generate approximately a quarter trillion in annual revenues.² CCIA members are substantially affected by the patent system and depend upon it to fulfill its constitutional purpose.

SUMMARY OF THE ARGUMENT

Software is unlike most types of technology in that the nature of software can easily result in the

¹ Pursuant to this Court's Rule 37.6, *amicus* affirms that no counsel for a party authored this brief in whole or in part, and no person other than the *amicus*, or its counsel, made a monetary contribution intended to fund the preparation or submission of this brief. The parties' letters consenting to the filing of this brief have been filed with the Clerk's office.

² A complete list of CCIA's members is available online at <http://www.ccianet.org/members>. CCIA is a trade association with a wide and diverse membership. It was unable to resolve the reservations of one of its members, Microsoft Corporation, prior to the filing of this brief. These views should therefore not be associated with Microsoft.

kind of overbroad patents that this Court has warned against since at least *O'Reilly v. Morse*, 56 U.S. 62 (1853). This is because software is inherently mathematical — that is, any piece of software code is computing a mathematical function — and because the lower courts have tended to allow software-related patents to become less and less tied to particular computer hardware.

This Court recognized in *Microsoft Corp. v. AT&T Corp.*, 127 S. Ct. 1746, 1755 (2007), that computer software that is not loaded onto particular computer hardware capable of executing that software is no more than an “idea.” While that case was not about patentable subject matter, the Court’s basic understanding of software was correct — computer software that is not tied to particular computer hardware is nothing more than an abstraction.

Moreover, a patent that includes software that is not limited to particular computer hardware is functional claiming, which this Court struck down in *Halliburton Oil Well Cementing Co. v. Walker*, 329 U.S. 1 (1946). Because the U.S. Court of Appeals for the Federal Circuit has created a simple, formalistic way to evade *Halliburton*, the problem of functional claiming that this Court sought to avoid has resurfaced.

This case offers the Court a chance to reaffirm *Halliburton* while giving clear guidance as to when software is patentable subject matter. *Amicus curiae* CCIA suggests that the key is the Court’s *Microsoft* decision, and tying software to *specific* hardware.

Accordingly, *amicus curiae* CCIA proposes a test for patent-eligibility of software that restricts

software to particular (as opposed to generic) computer hardware:

Is the software portion of the claim restricted to specific hardware, i.e., a particular type or architecture of computer hardware?

If so, the claim is patentable subject matter.

If not, treat the software portion of the claim as prior art³ and examine the remaining claim to determine if what is left is patentable subject matter.

This test (hereinafter, “the specific hardware test”) is supported by this Court’s prior decisions and is administrable. The specific hardware test also allows some discretion, in that the level of detail that must be specified for the required computer hardware is necessarily determined case-by-case. Regardless, a generic “computer” is never sufficient to make a software claim patent-eligible.

Under current Federal Circuit law, the patent-eligibility of software-related inventions has become controversial, with some commentators suggesting that allowing such patents at all has a negative impact on innovation.⁴ CCIA shares those commentators’ concerns.

³ See *Parker v. Flook*, 437 U.S. 586, 592 (1978).

⁴ See, e.g., Bronwyn H. Hall & Dietmar Harhoff, *Recent Research on the Economics of Patents*, NATIONAL BUREAU OF ECONOMIC RESEARCH, Working Paper 17773 (Jan. 2012), available at

In the interests of finding a practical solution, however, CCIA is proposing what it believes to be a reasonable, workable test that is consistent with this Court's precedent. Accordingly, CCIA respectfully requests that the Court adopt the specific hardware test in determining the patent-eligibility of software-related inventions.

ARGUMENT

I. AS THIS COURT RECOGNIZED IN *MICROSOFT V. AT&T*, SOFTWARE IS ONLY MEANINGFUL WHEN IT RUNS ON PARTICULAR COMPUTER HARDWARE

This Court has held for over a century and a half that an idea is not patentable. *Burr v. Duryee*, 68 U.S. 531 (1863). This Court also recently observed that “[a]bstract software code is an idea without physical embodiment...”⁵ Even though *Microsoft* did not address patentable subject matter, the observation is a general principle. Consistent with this principle, this Court should hold that software not tied to any particular computer hardware is not patent-eligible.

Terms such as “computer,” “network,” and “Internet website” are generic terms. There are literally hundreds of different types of computers, many different types of networks, and millions of types of Internet websites.

http://elsa.berkeley.edu/~bhhall/papers/HallHarhoff12_NBER_w17773.pdf.

⁵ *Microsoft*, 127 S. Ct. at 1755.

The result is that a patent claim that recites software but with no more specific hardware than a “computer” or a “website” is a purely functional claim. In a functional claim, at least one claim element is defined by the function it performs rather than a physical description. In a purely functional claim, the claim preempts every possible way of practicing the described function.

This Court recognized in *Halliburton* that functional claiming is improper, precisely because it grants an inventor too broad a monopoly. In the 1952 Patent Act, Congress created a very limited form of functional claiming, namely means-plus-function claiming. Means-plus-function claims allow a patentee to use functional claim language, but the scope of the functional claim is restricted to what the inventor describes in the specification and reasonable equivalents. Thus, means-plus-function claims avoid the problem of functional claiming (i.e., overbreadth) while still providing the convenience of describing elements by function.

Unfortunately, the U.S. Court of Appeals for the Federal Circuit has made a muddle of means-plus-function jurisprudence, turning what was supposed to be a limited exception to *Halliburton* into an easily circumvented formality. Essentially, the Federal Circuit has created a de facto rule that a claim is interpreted as means-plus-function if and only if it uses the term “means.” At the same time, the Federal Circuit places no limits on functional claims. Thus, an inventor is able to take advantage of the inherent overbreadth of functional claiming simply by using placeholder words other than “means.”

Accordingly, this Court should reaffirm its decision in *Halliburton* and disallow functional claims, except for the narrow exception created by Congress in 35 U.S.C. § 112(f).

A. Patents That Include Software But Are Not Restricted to Specific Computer Hardware Result in Overbroad Functional Claims

Functional claiming enables patent owners to blur claim boundaries and broadly claim technology they did not actually invent, taxing future players in the field. Functional claims are patent claims not directed to a particular machine or particular series of steps for achieving a goal, but rather the goal itself.

In the software context a functional claim means that the patentee claims to own the general function of his program, not merely the particular way he achieved that goal. Thus, the patentee has effectively captured, not just the particular instance of software he built, but also any other software that solves the same problem, no matter how different it is. The patentee claims to own the function itself.

This sort of unrestrained functional claiming has the potential to paralyze an industry.

A historical example illustrates this danger. The Wright Brothers, in their pioneering patent, did not claim the specific rudder-wing combination they invented, but rather the actual function of moving

the wing and the rudder simultaneously together.⁶ This broad patent allowed the Wright Brothers to clear the market of competitors, including those with far superior wing designs. It was not until the government forced the Wrights to license their patents in 1917 that airplane innovation finally took flight.⁷

This danger of functional claiming is still great in the modern era, as more complex technology has evolved. In the past few decades, the patent system has evolved such that even relatively simple products incorporate a combination of different components, each of which is often subject to multiple patents.

B. The Federal Circuit Has Created an Easy Way for Patentees to Evade *Halliburton* and Enforce Functional Claims

In *Halliburton*, the plaintiff challenged a patent drafted entirely in functional terms, referring to “means for” performing a series of functions.⁸

⁶ U.S. Patent No. 821,393, claim 7 (filed Mar. 23, 1903) (“means for simultaneously moving the lateral portions [of a wing] into different angular relations” and “means whereby said rudder is caused to present to the wind that side thereof . . . having the smaller angle of incidence”).

⁷ See Walter J. Boyne, *The Wright Brothers: The Other Side of the Coin*, WINGS OVER KANSAS, Apr. 20, 2008, available at <http://www.wingsoverkansas.com/boyne/a268>.

⁸ *Halliburton*, 329 U.S. at 8–9.

Striking down “broadness, ambiguity, and [the] overhanging threat of the functional claim,” the Court held that the claims were indefinite because they did not specify how the patent performed the function or limit the invention to the particular means that the patentee had actually invented.⁹ Exploiting “conventional functional language at the exact point of novelty,” the Court said, was not commensurate with what the patentee was removing from the public domain.¹⁰

When Congress passed the Patent Act of 1952 it created a narrow exception to the *Halliburton* decision to permit limited functional claiming, subject to certain conditions. Specifically, 35 U.S.C. § 112(f) (originally codified as 35 U.S.C. § 112, ¶ 6) allowed functional claims only if they were limited to the technologies detailed in the specification and reasonable equivalents. This allowed patentees the ability to use the convenient language of functional claiming without being able to claim more than they invented.

In recent years, however, the Federal Circuit has narrowly interpreted means-plus-function claims and opened the door to almost unrestricted functional claims. Interpreting Section 112(f) myopically, the Federal Circuit has said that the use of the term “means” creates a presumption that a claim element is a means-plus-element, and the absence of that

⁹ *Id.* at 12-13.

¹⁰ *Id.* at 10.

term creates the opposite presumption.¹¹ In practice, however, these “presumptions” have become the rule.¹² The result has been evasive patent drafting: patentees have circumvented the limits of Section 112(f) by simply avoiding usage of the term “means,” resulting in a dramatic increase in broad functional claims similar to the lack of structure in the pre-*Halliburton* era.¹³

Instead of requiring patentees to disclose the structure of the underlying hardware, courts have allowed this requirement to be fulfilled by generic language like “computer” or “data processing system.” For example, lower courts have found structure in language such as “a computer programmed to perform X” or “a computer aided method.”¹⁴ The term “computer,” however, does not define any particular architecture or device; it is a generic term encompassing literally thousands of types of machines. Thus, patentees can currently circumvent Section 112(f) by adding structure through generic hardware limitations that ultimately

¹¹ See, e.g., *York Prods., Inc. v. Central Tractor Farm & Family Ctr.*, 99 F.3d 1568, 1574 (Fed. Cir. 1996).

¹² See Mark A. Lemley, *Software Patents and the Return of Functional Claiming*, 2013 WIS. L. REV. 905, 944-45 (2013).

¹³ See *id.*

¹⁴ See *DealerTrack, Inc. v. Huber*, 674 F.3d 1315, 1331 (Fed. Cir. 2012); *Renhcol Inc. v. Don Best Sports*, 548 F. Supp. 2d 356, 362 (E.D. Tex. 2008).

preempt use of the software across multiple technologies, including personal computers, phones, and tablets.

The result has been rampant functional claiming. Professor Dennis Crouch, in a study of all utility patents released between 1976 and 2011, found that use of means-plus-function language in independent claims has dropped from over 60% to less than 10% in the past 25 years.¹⁵ In its stead, functional claiming outside the scope of Section 112(f) has grown. In that same period, independent claims incorporating functional language without actually using the word “means”—for example, “for calculating” or “for computing”—have nearly doubled from around 25% to almost 50%. Furthermore, independent claims incorporating the language referencing computers “configured to” or “configured for,” language commonly used in functionally claimed patents, have grown from close to 0% to almost 30%. This troubling trend reflects how patentees have reacted to the Federal Circuit’s and lower courts’ rulings, abandoning the means-plus-function claiming specified by Section 112(f) for effectively equivalent language without the crucial limitations.

In failing to restrict the unchecked growth of functional claiming, the Federal Circuit has neglected the root of ambiguous patents that include

¹⁵ Dennis Crouch, *Functional Claim Language in Issued Patents*, PATENTLY-O, Jan. 23, 2014, available at <http://patentlyo.com/patent/2014/01/functional-language-patents.html>.

software. This Court can end the abuse of Section 112(f) by reaffirming the standards from *Halliburton* and the 1952 Patent Act to guide the lower courts.

II. THIS COURT SHOULD ADOPT A TEST THAT REQUIRES SOFTWARE TO BE RESTRICTED TO A PARTICULAR COMPUTER ARCHITECTURE OR AN INDEPENDENTLY PATENTABLE PROCESS OR MACHINE

Amicus curiae CCIA proposes the following test for determining whether a software claim (or other computer-implemented invention) recites patentable subject matter under 35 U.S.C. § 101:

Is the software portion of the claim restricted to specific hardware, i.e., a particular model or models of computer hardware?

If so, the claim is patentable subject matter.

If not, treat the software portion of the claim as prior art¹⁶ and examine the remaining claim to determine if what is left is patentable subject matter.

This “specific hardware test” is based on this Court’s prior decisions. The specific hardware test simplifies the problem of multiple levels of abstraction that typically occurs in software, while still allowing for patents that include software. The

¹⁶ See *Parker v. Flook*, 437 U.S. 586, 592 (1978).

specific hardware test also prevents overbroad functional claiming.

A. If Software in a Patent Claim Is Restricted to a Particular Computer Architecture, It Is Not Abstract

The main confusion about the patentability of software is whether adding a “computer” or other generic device to a claim is sufficient to make the claim patent-eligible. This confusion has generally sprung from the Federal Circuit’s decision in *In re Alappat*, which states that “such programming creates a new machine, because a general purpose computer in effect becomes a special purpose computer once it is programmed to perform particular functions pursuant to instructions from program software.”¹⁷

While this statement is true as a vague principle, any particular piece of software code can only be installed on particular types of computers. For example, a word processing program might run on computers that run some versions of Microsoft Windows, but not on computers that run the Linux or Mac OS operating systems, because the same software code will not run on different operating systems. And certainly, it would make no sense to consider playing a video game on the computer in a car that controls the engine; nor would it make sense to consider the software program that runs a factory

¹⁷ *In re Alappat*, 33 F.3d 1526, 1545 (Fed. Cir. 1994) (en banc).

assembly line being executed on a computer that controls a microwave oven.

The generic term “computer” includes a wide variety of devices with an enormous range of capabilities, such as an abacus, a digital computer that uses vacuum tubes, a tablet computer, a traditional desktop computer, and an IBM mainframe computer that takes up an entire room. Computers vary enormously in power and storage, and not every computer is capable of performing any computable task.

But if a patent claiming software is tied to particular computer hardware, such as a particular type of desktop computer, a particular configuration of computer servers, or a computer system embedded in a device (like a car), it is not an abstract idea. Rather, it is a new machine.

How particular the patent claim must be in specifying computer hardware must be evaluated on a case-by-case basis. At a minimum, the particular computer hardware must have existed at the time the patent was filed and must have sufficient capability to execute the claimed software invention.

Some of the relevant characteristics of computer hardware include: processing speed, number of central processing units (CPUs), size of computer memory, non-volatile data storage, distribution of software components and/or data among computers connected by a network, operating system, programming language used, and/or required software that must be installed for the invention to work, e.g., a web browser, a particular word processing program, etc.

The Court may adopt additional considerations as well. But it is essential that these considerations support the goal of limiting patent claims to what the inventors actually invented.

B. Means-Plus-Function Form Can Limit a Claim to a Particular Computer Architecture

For software claims not expressly limited to a particular computer architecture, means-plus-function claiming offers an alternative way to restrict a software claim to particular computer hardware. Congress created means-plus-function claims to allow limited functional claiming:

An element in a claim for a combination may be expressed as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof, and such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof. 35 U.S.C. § 112(f).¹⁸

As this Court has recognized in *Parker v. Flook*, 437 U.S. 586 (1978), and *Diamond v. Diehr*, 450 U.S. 175 (1981), software without hardware is merely an abstract idea, and can only be saved through the structure that hardware limitations provide. These limitations can be contained in the

¹⁸ Formerly § 112, sixth paragraph, under the Patent Act of 1952. Renumbered to § 112(f) under the Leahy-Smith America Invents Act of 2011.

claims of the patent. However, software claims may invoke Section 112(f) through the use of means-plus-function language. In that situation, courts should look to the specification to find this “structure” in the hardware, and accordingly limit the claims.

C. If Software in a Patent Claim Is Not Limited to Particular Computer Hardware, This Court’s Precedent Says That the Software Cannot Be Considered When Assessing Patentability

If software recited in a patent claim is not limited to specific hardware, the software portion of the claims cannot be included for assessing patent-eligibility. The claim should be treated as if the software was well known.¹⁹ Then, the court should analyze the claim as a whole to see if it recites anything more than simply executing the software on generic computer hardware.²⁰

Computer hardware, separate from software, does not create meaningful limitations to software claims under Section 101. Though Petitioners claim that computers function “through the physical manipulation of switches” and thus “cannot be an abstract idea in the sense of a disembodied concept,” Brief for Petitioner at 35, this claim belies both the history and the modern reality of what defines a computer. The word “computer” first arose in the context of machines in the 19th century as a device or

¹⁹ *Flook*, 437 U.S. at 592.

²⁰ *Id.* at 596; *Mayo Collaborative v. Prometheus Labs.*, 132 S. Ct. 1289, 1294 (2012).

machine for performing or facilitating calculations, composed of a complex series of switches and gears. Thus, the earliest “computers” simply performed math. In the 1800s, the computer’s setup of gears and switches represented its hardware, while the mathematical functions performed represented its software. Of course, using these computers to perform abstract mathematical functions did not suddenly render these mathematical functions patentable. A specific machine’s hardware, with its unique setup of gears and switches, may have been patentable. But that would not have stopped another inventor from using a different machine, with an alternate setup of gears and switches, from performing the same mathematical function.

Computers have evolved, but the distinction between hardware and software remains the same. Computer software has expanded beyond merely performing math to a variety of applications, including speech recognition, operating systems, and financial services. As a result, the question of patentability has become more blurred than it was in the 19th century.

The usefulness of Section 101 to determine the patentability of software is blunted by the Federal Circuit’s reliance on unrelated hardware limitations. When patentees recite a generic “computer” in their claims, they are using insufficient hardware limitations to save software from abstraction. For example, Judge Lourie found in *Accenture v. Guidewire* that “simply implementing an abstract concept on a computer, without meaningful

limitations to that concept, does not transform a patent ineligible claim into a patent-eligible one.”²¹ Furthermore, using a “generic computer” to make a software claim patent-eligible is contrary to this Court’s ruling in *Flook*. The abstraction doctrine “cannot be circumvented by attempting to limit the use of the [abstract principle] to a particular technological environment.”²² By focusing the Section 101 inquiry on hardware instead of software, patentees are shielding the software space from the abstraction doctrine.

Petitioner²³ and at least one *amicus*²⁴ acknowledge that hardware and software are inseparable, and that each instruction in the software inevitably leads to a physical change in circuitry of the hardware. And this Court observed in *Microsoft* that software that is not tied to computer hardware is nothing more than an “idea.”²⁵

But Petitioner fails to note the discrepancy between the treatment of hardware and software within the patent system. While the USPTO and courts require hardware patentees to disclose specific

²¹ *Accenture Global Servs. v. Guidewire Software*, 728 F.3d 1336, 1345 (Fed. Cir. 2013) (citing *Bancorp Servs. v. Sun Life Assur. Co. of Canada*, 687 F.3d 1266, 1280 (Fed. Cir. 2012)).

²² *Flook*, 437 U.S. at 584.

²³ Pet. Br. at 37.

²⁴ Br. of *amicus curiae* IEEE at 9.

²⁵ *Microsoft*, 127 S. Ct. at 1755.

circuitry—resistors, transistors, gates, power sources—to gain patent protection, there is no similar requirement for software-related patents to disclose a specific implementation of an algorithm. Thus, a patentee can claim not just the software he invented, but the entire function itself. If hardware and software are one and the same, they should be held to the same standards of specificity.

III. THE PROPOSED TEST OFFERS A CLEAR SOLUTION THAT DEMONSTRATES THE INVALIDITY OF ALICE'S ASSERTED PATENT CLAIMS

The specific hardware test clearly demonstrates that Alice's claims are invalid under 35 U.S.C. § 101. Claim 26 of U.S. Pat. No. 7,725,375 is representative of the system claims:

A data processing system to enable the exchange of an obligation between parties, the system comprising:

a communications controller,

a first party device, coupled to said communications controller,

a data storage unit having stored therein

(a) information about a first account for a first party, independent from a second account maintained by a first exchange institution, and

(b) information about a third account for a second party, independent from a fourth account maintained by a second exchange institution; and

a computer, coupled to said data storage unit and said communications controller, that is configured to

(a) receive a transaction from said first party device via said communications controller;

(b) electronically adjust said first account and said third account in order to effect an exchange obligation arising from said transaction between said first party and said second party after ensuring that said first party and/or said second party have adequate value in said first account and/or said third account, respectively; and

(c) generate an instruction to said first exchange institution and/or said second exchange institution to adjust said second account and/or said fourth account in accordance with the adjustment of said first account and/or said third account, wherein said instruction being an irrevocable, time invariant obligation placed on said first exchange institution and/or said second exchange institution.

The specific hardware test first asks whether the claim is limited to specific hardware, either expressly or using means-plus-function format. Claim 26 recites several generic terms, namely a “computer,” a “communications controller,” a “first party device,” and a “data storage unit.” None of these are adequately specified in the claim itself. The claim contains no information about what type of computer is used, what type of communications controller is used, what type of data storage unit is

used, or any information at all about what a “first party device” is.

Instead, the claim defines these nouns in terms of the functions they perform. For example, the “data storage unit” is defined by what data it stores, and the “computer” is defined by what functions it is configured to perform.

Alice does not assert that this claim is a means-plus-function claim, so the claim is not limited to particular computer hardware.

According to the specific hardware test, the software portions of claim 26, i.e., the functions recited in the claim, must be treated as known. This leaves a computer connected to a communications controller and a data storage unit. Such a system is certainly not patent-eligible.

At best, the claim is for the idea of intermediated transactions. This idea is no more patentable than the idea of hedging risk that this Court rejected in *Bilski*.²⁶

Thus, Alice’s asserted system claims are not patent-eligible.

²⁶ *Bilski*, 130 S. Ct. at 3231.

CONCLUSION

For the foregoing reasons, *amicus curiae* CCIA asks this Court to adopt the specific hardware test for determining the patent-eligibility of software and find that Alice's asserted claims are not patentable subject matter under 35 U.S.C. § 101.

Respectfully submitted,

MATTHEW LEVY
Counsel of Record
ALEXANDRA STERNBURG
BRIAN KAHIN
COMPUTER &
COMMUNICATIONS
INDUSTRY ASSOCIATION
900 17th Street, NW,
Suite 1100
Washington, DC 20006
(202) 783-0070
mlevy@ccianet.org

February 27, 2014