

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

EBAY INC., ALIBABA.COM HONG KONG LTD., AND
BOOKING.COM B.V.,

Petitioner,

v.

GLOBAL EQUITY MANAGEMENT (SA) PTY. LTD.,

Patent Owner.

Case IPR2016-01829
Patent 7,356,677 B1

Before KARL D. EASTHOM, MATTHEW R. CLEMENTS, and
KEVIN C. TROCK, *Administrative Patent Judges*.

TROCK, *Administrative Patent Judge*.

FINAL WRITTEN DECISION and ORDER
35 U.S.C. § 318(a) and
37 C.F.R. § 42.73

I. INTRODUCTION

A. Background

eBay Inc., Alibaba.com Hong Kong Ltd., and Booking.com B.V. (“Petitioner”) filed a Petition (Paper 1, “Pet.”) to institute an *inter partes* review of claims 1–7 of U.S. Patent No. 7,356,677 B1 (Ex. 1001, “the ’677 patent”). Global Equity Management (SA) Pty. Ltd. (“Patent Owner”) filed a Preliminary Response to the Petition. Paper 10 (“Prelim. Resp.”). We instituted an *inter partes* review of claims 1, 2, 6, and 7 of the ’677 patent. Paper 13 (“Dec. to Inst.”). Patent Owner filed a Patent Owner Response (Paper 27, “PO Resp.”), Petitioner filed a Petitioner Reply (Paper 42, “Pet. Reply”), and a hearing was held on January 22, 2018, a transcript of which has been entered into the record (Paper 59, “Tr.”).

We have jurisdiction under 35 U.S.C. § 6(b). This Final Written Decision is issued pursuant to 35 U.S.C. § 318(a). We base our decision on the preponderance of the evidence. 35 U.S.C. § 316(e); 37 C.F.R. § 42.1(d). Having reviewed the arguments of the parties and the supporting evidence, we find that Petitioner has demonstrated by a preponderance of the evidence that claims 1, 2, 6, and 7 of the ’677 patent would have been obvious over the teachings of the prior art references.

B. The ’677 Patent

The ’677 patent describes the invention as

a method of managing a computer system [that] provides for rapid switching among operating systems (“OS”) of a computer having multiple installed operating systems. This inventive method and system manager device make special use of the prior art power management (“PM”) support functions of the runtime operating system and the basic input/output system (“BIOS”) to save a currently active operating system environment in a storage

device and prepare to turn off the computer system, or to switch to, execute, activate or initiate, another operating system as designated by the user. An operating system that is suspended by the PM may be referred to as being suspended, hibernating or in a suspended state. When the user wishes to activate a suspended (hibernating) operating system environment, he need not reboot the computer and start from the cold state. Instead, the user can switch back to the suspended operating system and quickly resume operation of operating system and applications, in a manner similar to turning on an appliance.

Ex. 1001, 6:39–57.

C. Challenged Claims of the '677 Patent

Challenged claims 1 and 6 are independent. Challenged claims 2 and 7 depend from claims 1 and 6, respectively. Claim 1 is illustrative and is reproduced below.

1. A hardware platform for a hibernate capable computer system comprising a system manager, said computer system having an OS-independent storage manager operating through a firmware level and a plurality of operating systems and applications, said storage manager having a virtual table of contents for organizing and accessing a plurality of partitions of relevant data and having a plurality of virtual computer systems, each of said virtual computer systems capable of accessing a selection of the partitions, the virtual table of contents (VTOC) being capable of dynamically configuring a plurality of partition tables, said system manager comprising:

means for selecting one of said virtual computer systems to become next operable before suspending a currently operational virtual computer system;

means for suspending the currently operational virtual computer system in an active state;

means for making the selected virtual computer system operable into a running state, and

means for switching of the virtual computer systems using a switch flag and BIOS ACPI solutions, and without initialization of power-on self test (POST) in the BIOS, wherein the switch flag is a flag that is set up in storage to differentiate between suspend for fast switching and power save suspend.

Ex. 1001, 16:5–29.

II. DISCUSSION

A. Claim Construction

In an *inter partes* review, claim terms in an unexpired patent, such as the '677 patent, are given their broadest reasonable construction in light of the specification of the patent. 37 C.F.R. § 42.100(b); *Cuozzo Speed Techs., LLC v. Lee*, 136 S. Ct. 2131, 2144–46 (2016) (upholding the use of the broadest reasonable construction as the standard to be applied for claim construction in *inter partes* reviews). Under this standard, and absent any special definitions, we give claim terms their ordinary and customary meaning, as would be understood by one of ordinary skill in the art at the time of the invention. *In re Translogic Tech., Inc.*, 504 F.3d 1249, 1257 (Fed. Cir. 2007). Moreover, only those terms that are in controversy need be construed, and only to the extent necessary to resolve the controversy. *Vivid Techs., Inc. v. Am. Sci. & Eng'g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999).

Petitioner identifies particular claim terms for construction along with proposed constructions. Pet. 14–26. Petitioner proposes that we adopt these constructions despite the parties' respective constructions submitted in *Global Equity Management (SA) Pty. Ltd. v. Expedia.com, et al.*, Case No.

2:16-cv-0095 (E.D. Tex.) (“Lead Case”¹). *Id.* at 15. Petitioner asserts that its proposed constructions should at least be encompassed within the broadest reasonable interpretation standard applicable here, and thus, should be adopted for purposes of this proceeding. *Id.*

In the “Claim Construction” section of its Response (Section V), Patent Owner neither responds directly to Petitioner’s proposed claim constructions nor proposes alternative constructions. *See* PO Resp. 13–14. Instead, Patent Owner acknowledges the district court’s constructions in the Lead Case and states, “[Patent Owner] notes that claim construction in *inter partes* review is broader than in litigation. Nevertheless, the constructions made by the District Court are instructive.” *Id.*

Petitioner asserts that Patent Owner ignored its constructions and left un rebutted the assertion that Petitioner’s constructions were reasonable. Pet. Reply 3. Petitioner argues that Patent Owner has waived any claim construction arguments. *Id.*

We determine that the following terms require express construction. We do not find it necessary to construe expressly additional claim terms for purposes of this decision.

1. “*virtual computer system*”

Independent claims 1 and 6 recite the phrase “virtual computer system.” Petitioner contends that during the prosecution of the ’677 patent the Examiner stated that this phrase was not a widely-used term in the art, and therefore, that its definition depended on the definition provided in the Applicant’s specification. Pet. 16 (citing Ex. 1004, 326). Based on the

¹ Petitioner identifies thirty-seven related cases (Pet. 8–12), but refers to Case No. 2:16-cv-0095 (E.D. Tex.) as the “Lead Case” (*id.* at iv; Paper 8, 1).

specification, Petitioner argues, the Examiner stated that the Applicant had defined “virtual computer system” or “virtual PC” as “a set of partitions and file systems that are visible and available to a given operating system environment when that environment is actively executing on the physical computer.” *Id.* Petitioner argues that the ’677 specification supports this construction. *Id.* (citing Ex. 1001, 7:29–32). During prosecution, Petitioner argues, the Applicant did not disagree with this construction. *Id.* Petitioner thus contends that the Examiner’s construction should apply in this proceeding under the applicable claim construction standard.

In Section VI of Patent Owner’s Response, Patent Owner points out that the district court in the Lead Case construed the term “virtual computer system” to mean a “set of partitions and file systems that are visible and available to a given operating system environment when that environment is actively executing on the physical computer, having a unique master boot record and its own partition table.” PO Resp. 22 (citing Ex. 1015, 68). Patent Owner points to a section of the ’677 patent discussing a master boot record (MBR), but provides no argument or explanation as to how this section of the ’677 patent relates to the district court’s construction of “virtual computer system.” PO Resp. 22–23 (quoting Ex. 1001, 8:56–65).

We note that district courts use a different legal standard than the Board in construing claim terms. *See Cuozzo Speed Techs., LLC v. Lee*, 136 S. Ct. at 2144–2146 (2016); *Novartis AG v. Noven Pharms. Inc.*, 853 F.3d 1289, 1293–94 (Fed. Cir. 2017).

Here, the ’677 patent provides an express definition of a “Virtual PC” or “Cabinet,” terms which appear to be used synonymously in the ’677 patent with the term “virtual computer system.” *See, e.g.*, Ex. 1001,

Abstract (“Combinations of compatible partitions can be assigned to a cabinet to create a virtual computer system”), 7:62–63 (“As used herein, the terms ‘Cabinet’ and ‘Virtual PC’ are interchangeable.”). The ’677 patent defines a “Virtual PC” or “Cabinet” as “a set of partitions and file systems that are visible and available to a given operating system environment when that environment is actively executing on the physical computer.” Ex. 1001, 7:30–32. One of ordinary skill in the art, therefore, in reading the ’677 patent, would understand the claim term “virtual computer system” to be synonymous with “Virtual PC” and “Cabinet” and mean “a set of partitions and file systems that are visible and available to a given operating system environment when that environment is actively executing on the physical computer.” This is the broadest reasonable interpretation of the term “virtual computer system,” given the express definition provided by the ’677 patent. Patent Owner’s proposed additions of a “unique master boot record” and “its own partition table” would create an unreasonably narrow construction given the express definition of “virtual computer system” in the ’677 patent.

2. “*virtual table of contents (VTOC)*”

Claims 1 and 6 recite the phrase “virtual table of contents” or “VTOC.” Petitioner contends that during the prosecution of the ’677 patent, the Examiner stated that the limitation “VTOC” was not a widely-used term in the art and therefore its definition depended on the definition provided in the specification. Pet. 17 (citing Ex. 1004, 326). Petitioner asserts that the Examiner found, based on the specification, that the appropriate definition of “virtual table of contents” or “VTOC” was a “data structure maintained on secondary storage, but outside the primary file systems” and “contains all

necessary information about [a] Virtual PC [*i.e.* the virtual computer system] of a physical computer, which includes storage information, partitions and file systems.” *Id.* Petitioner argues that the specification supports this construction. *Id.* (citing Ex. 1001, 7:23–28). Petitioner argues that during prosecution of the ’677 patent, Applicant did not disagree with this construction. *Id.*

Patent Owner notes that the ’677 specification is consistent with the district court’s claim construction in the Lead Case, which determined that “virtual table of contents” means a “data structure maintained on secondary storage, but outside the primary file systems, that contains all necessary information about each virtual computer system environment of a physical computer, including storage information, partitions and file systems.” PO Resp. 19 (quoting Ex. 1015, 67).

The parties, therefore, appear to agree on an appropriate construction for the term “virtual table of contents” or “VTOC.” Because the ’677 patent provides an express definition, we find that the broadest reasonable construction of “virtual table of contents” or “VTOC,” when read in light of the specification as it would be interpreted by one of ordinary skill in the art, is a “data structure maintained on secondary storage, but outside the primary file systems, which contains all necessary information about each virtual computer system environment of a physical computer, including storage information, partitions and file systems.”

3. “*means for selecting*”

Claim 1 recites “means for selecting one of said virtual computer systems to become next operable before suspending a currently operational virtual computer system.” Petitioner asserts that the recited function is

“selecting one of said virtual computer systems to become next operable before suspending a currently operational virtual computer system,” but asserts that there is no corresponding structure disclosed in the specification. Pet. 25. Petitioner, however, notes that Patent Owner has posited that the structure disclosed for at least one of the limitations related to “choosing” an operating system is a “Graphical user interface (GUI) or the like.” *Id.* (citing Ex. 1009, 34; Ex. 1001, 10:1–4).

Patent Owner points out that the district court in the Lead Case construed “means for selecting one of said virtual computer systems to become next operable before suspending a currently operational virtual computer system” under 35 U.S.C. § 112 ¶ 6, with the function of “selecting one of said virtual computer systems to become next operable before suspending a currently operational virtual computer system” and with a corresponding structure of the “Super OS GUI, and equivalents.” PO Resp. 25 (quoting Ex. 1015, 90).

We have considered the claim constructions proffered by the parties, and the claim construction issued by the district court in the Lead Case. We agree with the district court that the “means for” clause of this limitation falls within the ambit of 35 U.S.C. § 112 ¶ 6. We determine the function to be “selecting one of said virtual computer systems to become next operable before suspending a currently operational virtual computer system.” The ’677 patent explains that “the Super OS GUI provides a selection means for choosing one of the available operating systems and corresponding virtual compute systems.” Ex. 1001, 10:2–4. The ’677 patent also illustrates “a GUI for switching between the available Operating Systems in the computer,” and illustrates a similar GUI “to confirm the selection of a

particular Operating System environment.” See Ex. 1001, 6:21–24, Figs. 7, 8, 13:17–19. Accordingly, we determine the corresponding structure to be a “GUI, Super OS GUI, or equivalents.”

4. “*means for switching*”

Claim 1 recites “means for switching of the virtual computer systems using a switch flag and BIOS ACPI solutions, and without initialization of power-on self test (POST) in the BIOS.” Petitioner asserts that the recited function is “switching of the virtual computer systems using a switch flag and BIOS ACPI solutions, and without initialization of power-on self test (POST) in the BIOS.” Pet. 22. Petitioner argues that the structure for this term includes the following algorithm: 1) note the virtual computer system to be loaded next and set the switch flag; 2) suspend currently active OS environment; 3) proceed without initialization of power-on self test (POST) in the BIOS; 4) adjust MBR to reflect the newly activated virtual computer system; and 5) invoke BIOS resume procedure to restore the state of a previously suspended OS environment of the newly activated virtual computer system, and equivalents thereof. *Id.* at 23–24 (citing Ex. 1001, 10:10–54).

Patent Owner does not contest Petitioner’s proposed construction and does not propose any alternative construction for us to adopt. See PO Resp. 12–14, 30–33.

The district court in the Lead Case construed the limitation “means for switching of the virtual computer systems using a switch flag and BIOS ACPI solutions, and without initialization of power-on self test (POST) in the BIOS, wherein the switch flag is a flag that is set up in storage to differentiate between suspend for fast switching and power save suspend”

under 35 U.S.C. § 112 ¶ 6 as having the function of “switching of the virtual computer systems using a switch flag and BIOS ACPI solutions, and without initialization of power-on self test (POST) in the BIOS, wherein the switch flag is a flag that is set up in storage to differentiate between suspend for fast switching and power save suspend,” with the supporting structure identified in the ’677 patent at column 10, line 63 thru column 11, line 60 (“suspend”/ “restore” / “resume”), or column 12, lines 15–41 and column 11, line 48, and equivalents. Ex. 1015, 99.

The ’677 patent recites a specific algorithm performed by the Super OS program to switch virtual computer systems using a switch flag and BIOS ACPI solutions and without initialization of the power-on self test in the BIOS. Ex. 1001, 10:10–54. The steps are:

- 1) Note the virtual computer system to be loaded next and set the switch flag. Ex. 1001, 10:15–17;
- 2) Suspend currently active OS environment. *Id.* at 10:18–20;
- 3) Proceed without initialization of power-on self test (POST) in the BIOS. *Id.* at 10:22–27, 4:36–43;
- 4) Adjust MBR (master boot record) to reflect the newly activated virtual computer system. *Id.* at 10:35–36; and
- 5) Invoke BIOS resume procedure to restore the state of a previously suspended OS environment of the newly activated virtual computer system. *Id.* at 10:37–41.

We agree with the district court that the “means for” clause of this limitation falls under 35 U.S.C. § 112 ¶ 6. We determine the function to be “switching of the virtual computer systems using a switch flag and BIOS ACPI solutions, and without initialization of power-on self test (POST) in

the BIOS.” Because the ’677 patent provides a specific algorithm to accomplish this function, we determine the corresponding structure to be a processor performing the algorithm: 1) note the virtual computer system to be loaded next and set the switch flag; 2) suspend the currently active OS environment; 3) proceed without initialization of power-on self test (POST) in the BIOS; 4) adjust MBR to reflect the newly activated virtual computer system; and 5) invoke the BIOS resume procedure to restore the state of a previously suspended OS environment of the newly activated virtual computer system, and equivalents thereof.

5. “*fast resuming*”

Dependent claim 2 recites, “[t]he hardware platform as in claim 1, wherein: said means for making the selected virtual computer system operable comprising means for fast resuming said selected virtual computer system.” Dependent claim 7 recites, “[t]he method as in claim 6, wherein said activating step further comprises: fast resuming said another virtual computer system to a current state.”

Petitioner contends that, “fast resuming” should be construed to mean making modified use of the BIOS power management standard to allow transfer from suspended state to operating state without POST and without loading all components of an operating system from the cold boot state. Pet. 24–25. Petitioner asserts the ’677 patent describes an algorithm performed by the Super OS program to fast resume a selected virtual computer system. *Id.* at 24 (citing Ex. 1001, 10:22–54). Petitioner asserts that the steps to the algorithm are:

1) Proceed without initialization of power-on self test (POST) in the BIOS. *Id.* (citing Ex. 1001, 10:22–27, 4:36–43);

2) Adjust MBR (master boot record) to reflect the newly activated virtual computer system. *Id.* (citing Ex. 1001, 10:35–36); and

3) Invoke BIOS resume procedure to restore the state of a previously suspended OS environment of the newly activated virtual computer system.

Id. at 24–25 (citing Ex. 1001, 10:37–41, 10:35–36).

Petitioner explains that using the BIOS resume procedure will transfer the computer system into an operating state without loading all components of an operating system from the cold boot state. Pet. 25 (citing Ex. 1001, 6:52–57).

Patent Owner neither contests Petitioner’s proposed construction nor provides any alternative construction for us to consider.

Because the ’677 patent describes how to transfer a computer system from a suspended state into an operating state without loading all components of an operating system from the cold boot state, we construe “fast resume” to mean “making modified use of the BIOS power management standard to allow transfer from suspended state to operating state without POST and without loading all components of an operating system from the cold boot state.”

B. Level of Ordinary Skill in the Art

In determining whether an invention would have been obvious at the time it was made, we consider the level of ordinary skill in the pertinent art at the time of the invention. *Graham v. John Deere Co.*, 383 U.S. 1, 17 (1966). “The importance of resolving the level of ordinary skill in the art

lies in the necessity of maintaining objectivity in the obviousness inquiry.” *Ryko Mfg. Co. v. Nu-Star, Inc.*, 950 F.2d 714, 718 (Fed. Cir. 1991).

Petitioner argues that a person of ordinary skill in the art (“POSITA”) relevant to the ’677 patent would have had a bachelor’s degree in computer science, computer engineering, or the equivalent, and at least two years’ experience in computer operating systems, programs and databases, and/or graphical user interfaces. Pet. 14 (citing Ex. 1002 ¶ 25).

In the Patent Owner Response, Patent Owner takes no position on the appropriate level of ordinary skill in the art. Patent Owner’s Declarant, Mr. Rafizadeh, opines that a person of ordinary skill in the art would have had a “a B.S. degree (or equivalent) in computer engineering or computer science, and four years of experience in virtualization and the design of GUI’s in the field of virtualization.” See Ex. 2017 ¶ 44. This paragraph, however, is neither cited nor argued by Patent Owner in its Response. Mr. Rafizadeh is the sole named inventor on the ’677 patent, and is one of two Directors of GEMSA, the Patent Owner, in this case. Ex. 1026, 104:10–14. Mr. Rafizadeh admits that he has a financial interest in GEMSA, (*id.* at 104:15–16), and that he has a personal and a financial interest in maintaining the validity of the ’677 patent (*id.* at 110:8–12).

“In assessing the probative value of an expert opinion, [the Board] must consider three factors: (1) the interest in the outcome of the case, (2) the presence or absence of factual support for the expert’s opinion, and (3) the strength of any opposing evidence.” *EC Data Sys., Inc. v. J2 Global, Inc.*, No. 2014-008203, 2014 WL 7171966, at *4 (PTAB Dec. 15, 2014) (citing *Ashland Oil, Inc. v. Delta Resins & Refractories, Inc.*, 776 F.2d 281 (Fed. Cir. 1985)); see also *Ex Parte Magna Elecs., Inc.*, No. 2013-004164,

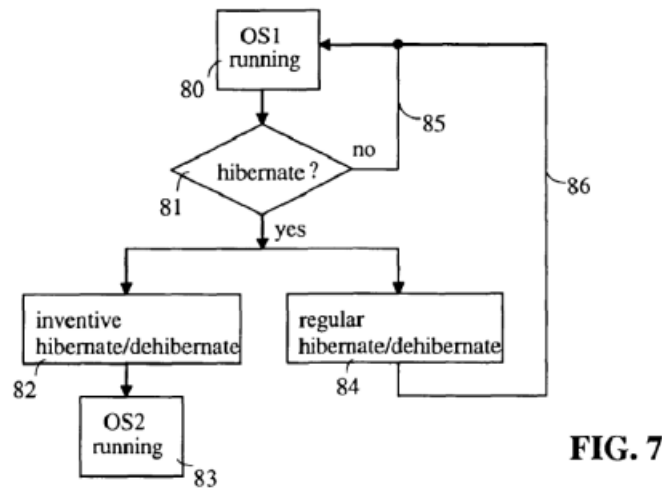
2014 WL 2360424, at *3 (PTAB May 28, 2014) (expert who “was employed by Patent owner when he provided the expert opinion . . . has a substantial interest in the outcome of the case”). Given Mr. Rafizadeh’s admitted personal and financial interest in the outcome of this case, we give his testimony less weight.

We therefore find that a person of ordinary skill in the art relevant to the ’677 patent would have had a bachelor’s degree in computer science, computer engineering, or the equivalent, and between about two to four years’ experience in computer operating systems, programs and databases, and/or graphical user interfaces or virtualization.

C. Hermann (Ex. 1005)

Hermann, a published European patent application, EP 1 037 133 A1, bears a publication date of September 20, 2000. Ex. 1005, 1. Patent Owner does not contest this publication date. Accordingly, Hermann qualifies as prior art to the ’677 patent under 35 U.S.C. §102(b). Hermann teaches a method for switching between two operating systems in a computer. Ex. 1005, Title, 2:38–47. Hermann accomplishes this by replacing the traditional shut-down/re-boot procedure normally required to switch operating systems with suspend and resume operations. *Id.* at 9:41–52. For the suspend operation, Hermann teaches saving the state of the currently running OS as a hibernation image. *Id.* at 9:35–37. Hermann teaches that the suspend operation may be followed by a resume operation when a user indicates a different saved hibernation image to be restored upon hibernation of the currently running OS. *Id.* at 10:30–37.

Figure 7 of Hermann is a flow chart illustrating a switch from a first operating system (OS1) to a second operating system (OS2). Figure 7 of Hermann is shown below.



Ex. 1005, Fig. 7. Figure 7 shows Hermann’s protocol for switching operating systems involving first hibernating the currently running operating system, OS1, and then dehibernating a second operating system, OS2, instance stored in a hibernation image. *Id.* at Fig. 7 (block 82). Hermann teaches that the switching of operating systems is accomplished “without having to shut-down and re-boot the system” or invoking a “power-down cycle.” *Id.* at 12:39–44, 16:30–33.

D. PowerQuest BootMagic User Guide (Ex. 1007)

PowerQuest BootMagic was a software product sold by PowerQuest Corp. that allowed users to run multiple operating systems on a single PC. Ex. 1007, xiv; Ex. 1011 ¶¶ 4, 7. The BootMagic User Guide was a software manual included with BootMagic software. *Id.* at A; Ex. 1011 ¶ 4.²

² Patent Owner has moved to exclude the BootMagic User Guide under Fed.

The BootMagic User Guide illustrates a graphical user interface that allowed users to select among multiple operating systems installed on a PC. Ex. 1007, xiv. Figure 0.1 from the BootMagic User Guide is shown below.

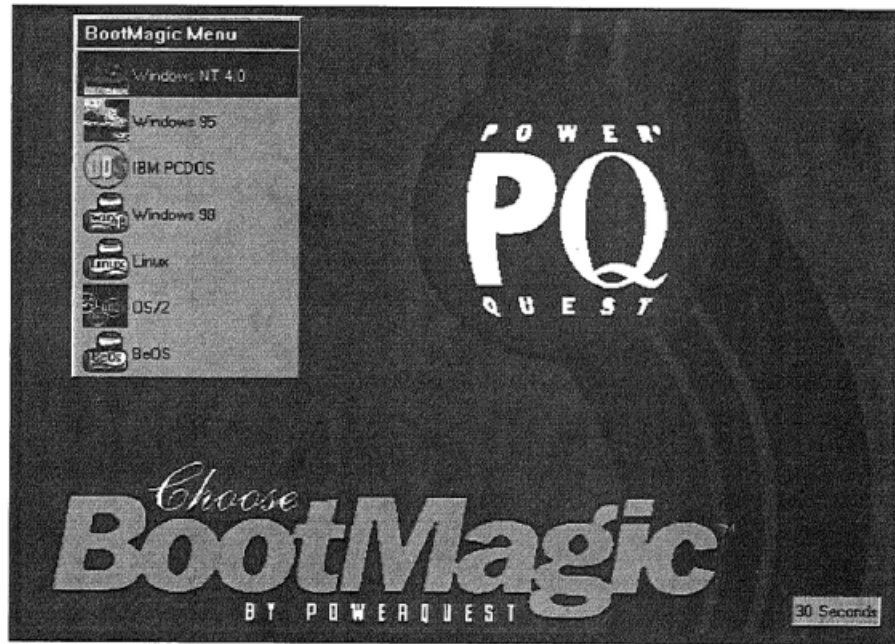


Fig. 0.1 BootMagic startup menu

Id. Figure 0.1 shows a screenshot of BootMagic’s graphical user interface. According to the BootMagic User Guide, once the user chooses a desired OS, the BootMagic software loads the OS’s boot record into memory, hides and unhides necessary partitions, and marks the OS’s associated partition as active. *Id.* at xv. The BootMagic software then launches the selected operating system. *Id.*

E. ACPI Specification (Ex. 1006)

The Advanced Configuration and Power Interface Specification, Rev.

R. Evid. 401–403, and 901 (Paper 48, 6–8), as well as the Ruff Declaration under 401–403, 602, 801–802, and 805 (*id.* at 3–5). We have addressed separately and denied Patent Owner’s motion to exclude the BootMagic User Guide and the Ruff Declaration.

1.0b, Feb. 2, 1999 (“ACPI Specification”) provides specifications for various computer system power states and transitions between those power states. Ex. 1006, 1, 30.

The ACPI Specification evolved from an existing collection of code and specifications, including the prior Advanced Power Management (“APM”) Specification. Ex. 1006, 13. The ACPI Specification defines several states that allow computers to conserve energy. *Id.* at 28. It also illustrates the transition between the working state and sleeping states. *Id.* at 171. A stated goal of the ACPI Specification is wide adoption to encourage hardware and software vendors to build ACPI-compatible implementations. *Id.* at 13.

F. Analysis

Petitioner contends claims 1, 2, 6, and 7 of the ’677 patent would have been obvious over the combination of Hermann, the BootMagic User Guide, and the ACPI Specification. Pet. 27–75. Claims 1 and 6 are independent, and claims 2 and 7 depend from them, respectively. Claim 6, which claims a method of managing a computer system, has similar limitations to claim 1, which claims a hardware platform for a hibernate capable computer system.

Patent Owner contends the references cannot be combined as alleged, and do not teach or suggest “virtual computer systems,” “virtual table of contents (VTOC) being capable of dynamically configuring a plurality of partition tables,” “means for selecting one of said virtual computer systems to become next operable before suspending a currently operational virtual computer system,” “means for switching of the virtual computer systems using a switch flag and BIOS ACPI solutions, and without initialization of power-on self test (POST) in the BIOS,” and “wherein the switch flag is a

flag that is set up in storage to differentiate between suspend for fast switching and power save suspend,” as recited in claim 1. PO Resp. 17–36. Patent Owner contends similar arguments apply to the patentability of claim 6. *Id.* at 14. Patent Owner relies on the Declaration of Schumann Rafizadeh (Ex. 2017) to support its position. Patent Owner does not address separately the patentability of dependent claims 2 and 7.

Having considered all the evidence provided and the arguments advanced by Petitioner and Patent Owner, we conclude that Petitioner has established by a preponderance of the evidence that claims 1, 2, 6, and 7 would have been obvious over the combination of Hermann, the BootMagic User Guide, and the ACPI Specification. We address the evidence and arguments presented by the parties and highlight specific issues in further detail below.

1. Independent Claim 1

Petitioner relies on Hermann to teach the recited “hardware platform for a hibernate capable computer system” of claim 1. Pet. 38; Ex. 1002 ¶ 88. Figure 5 of Hermann illustrates a computer hardware platform including a CPU, memory controller, main memory, ROM memory, disk drive, a data bus, and various peripherals. Ex. 1005, Fig. 5; Pet. 38. Hermann explains that the hardware platform is hibernate capable. Pet. 38; Ex. 1005, 2:51–57.

Petitioner relies on Hermann to teach the recited “computer system having an OS-independent storage manager operating through a firmware level and a plurality of operating systems and applications.” Pet. 39–42. Hermann teaches a computer having modified BIOS code, a custom boot loader, or a combination of the two, which, Petitioner argues, teaches an OS-independent storage manager. *Id.* at 39; see Ex. 1005, Fig. 5 (showing

modified BIOS code 58 within the computer system). Petitioner asserts the modified BIOS code disclosed by Hermann is independent of any of the operating systems contained in the computer system because the BIOS code does not depend on any particular operating system to execute. *Id.*; Ex. 1010, 2. The APM Specification, which Hermann incorporates by reference (see Ex. 1005, 1:30–39), illustrates that the APM BIOS code is OS independent. Ex. 1002 ¶ 91.

Petitioner argues that the modified BIOS code loads and executes before any operating system is running and therefore the modified BIOS code is running independent of any operating system. Pet. 39; see Ex. 1005, 7:13–32; Ex. 1002 ¶¶ 60, 91. Petitioner argues the modified BIOS code is also a storage manager because it manages the hibernation images located on disk drive storage via control information 71 and manages which partition on the hard drive is made “active.” Pet. 39; Ex. 1005, 7:51–54; Ex. 1002 ¶¶ 71–73, 92. Petitioner explains the modified BIOS code allows a user to choose a hibernation file to dehibernate and manages how the user saves hibernation files, e.g., to replace an existing file or to create a new file. *Id.*; Ex. 1005, 13:43–14:29.

Petitioner asserts that BIOS code is firmware code and therefore the modified BIOS code operates through a firmware level. Pet. 40; Ex. 1002 ¶ 93. In addition, Petitioner asserts that modified BIOS code operates through the plurality of operating systems and applications by invoking the Advanced Power Management (APM) suspend and resume functions. Pet. 40; Ex. 1005, 1:37–39.

Petitioner asserts the custom boot loader taught by Hermann is also independent of any of the operating systems contained in the computer

system since the custom boot loader code also does not depend on any particular operating system to execute. Pet. 40; Ex. 1005, 14:30–38, 7:45–49; Ex. 1002 ¶¶ 62, 91. Petitioner also asserts the custom boot loader operates through a firmware level because the custom boot loader operates through the BIOS level to carry out its functions. *Id.*; Ex. 1005, 15:5–17; Ex. 1002 ¶ 93.

Further, Petitioner argues, the custom boot loader operates through a plurality of operating systems and applications by providing an entry point to the operating system for the suspend and resume functions. Pet. 40–41; Ex. 1005, 14:38–40, 15:2–4; Ex. 1002 ¶ 63. Petitioner also argues the custom boot loader is a storage manager because it can also manage which partition on the hard drive is made “active.” Pet. 41; Ex. 1005, 7:51–54. Petitioner explains the custom boot loader has similar abilities to the modified BIOS code in management of hibernation files on a hard disk. Pet. 41; Ex. 1005, 14:47–55, 15:8–12.

Petitioner argues that because the modified BIOS code and the custom boot loader are OS-independent storage managers, a combination of these two items would also constitute an OS independent storage manager. Pet. 41.

Petitioner relies on Hermann to teach the recited “storage manager having a virtual table of contents for organizing and accessing a plurality of partitions of relevant data and having a plurality of virtual computer systems.” Pet. 42–46. Petitioner argues that because Hermann uses a computer system running both the OS/2 and Linux operating systems, it teaches two or more sets of partitions and file systems. *Id.* at 42; Ex. 1001, 11:38–50. Petitioner argues a person of ordinary skill in the art would

understand that the OS/2 operating system would necessarily have partitions and file systems separate from the Linux operating system. *Id.*; Ex. 1002 ¶¶ 36–38, 96; *see also* Ex. 1005, 7:49–51.

Petitioner argues Hermann teaches a virtual table of contents in the form of hibernation images and corresponding control information, depicted in Figure 6 of Hermann. Pet. 43; Ex. 1005, Fig. 6. Petitioner asserts the data shown in Figure 6 of Hermann constitutes a data structure because the data is organized into different hibernation images, labelled 70, and within each hibernation image the data is organized into data blocks consisting of file allocation information, work data required for hibernation, content of volatile memory, and the content of main memory. Pet. 43–44; Ex. 1005, 16:43–17:11.

Petitioner argues the data in Figure 6 contains all necessary information about the virtual computer system of a physical computer including storage information, partitions, and file systems. *Id.* at 44. The hibernation images, labelled 70 in Figure 6, contains all the necessary state information required to reestablish a respective OS instance and its corresponding processor states. Ex. 1005, 3:28–31; Pet. 44.

Petitioner argues that the data in Figure 6 is for organizing and accessing a plurality of partitions of relevant data, wherein each hibernation image in Figure 6 represents an OS instance that could be reestablished. Pet. 45; Ex 1005, 1:55–2:4, 9:35–37, 17:11–18. Once reestablished, the OS instance is used to access a set of partitions having data relevant to that OS instance. *Id.* at 46; Ex. 1002 ¶ 98. Therefore, Petitioner argues, the hibernation images, which are a part of the virtual table of contents, are used

for accessing certain partitions of data depending on which hibernation image is reestablished. Pet. 46; Ex. 1002 ¶ 98.

Petitioner relies on Hermann to teach the recited “means for suspending the currently operational virtual computer system in an active state.” Pet. 51–52. Hermann teaches APM (Advanced Power Management) BIOS routines such as the “suspend routine” for suspending the current OS in an active state. Ex. 1005, 9:12–16; Pet. 52. Hermann teaches that the suspend routine suspends the normal operating state and places the system into the suspend state. *Id.* at Fig. 2, 9:19–23; Pet. 52. Petitioner argues that the suspend routine is a power management support function. Pet. 52; Ex. 1002 ¶¶ 45–47. Hermann teaches that the suspend routine is also a part of the BIOS. Ex. 1005, 9:12–16; Pet. 52. Petitioner asserts the suspend routine is used to during the OS switching process. *Id.* at Fig. 7 (box 82), 1:51–56; Pet. 52.

Petitioner also relies on Hermann to teach the recited “means for making the selected virtual computer system operable into a running state.” Pet. 52–53. Hermann teaches the “resume” APM (Advanced Power Management) BIOS routine for making the selected OS operable into a running state. *Id.* at 52; Ex. 1005, 9:12–16. Petitioner argues that the resume routine resumes a suspend state, e.g. as stored in a hibernation file, into the normal operating state, i.e. the running state. *Id.*; see Ex. 1005, Fig. 2, 13:50–53. Petitioner argues the resume routine is a power management support function. *Id.* at 53; Ex. 1002 ¶ 109. Hermann teaches the resume routine is also a part of the BIOS. Ex. 1005, 9:13–16; Pet. 53. Petitioner asserts the resume routine is used to dehibernate a hibernation file during the OS switching process. *Id.* at Fig. 7 (box 82), 1:58–2:4; Pet. 53.

Notwithstanding Patent Owner's specific arguments, which we address below, we are persuaded by a preponderance of the evidence that the combination of Hermann, the BootMagic User Guide, and the ACPI Specification teaches or suggests the following limitations recited in claim 1:

[a] hardware platform for a hibernate capable computer system comprising a system manager, said computer system having an OS-independent storage manager operating through a firmware level and a plurality of operating systems and applications, said storage manager having a virtual table of contents for organizing and accessing a plurality of partitions of relevant data;

means for suspending the currently operational virtual computer system in an active state; and

means for making the selected virtual computer system operable into a running state.

Patent Owner contends, however, that the references do not teach or suggest certain other limitations recited in claim 1. We consider each of these contested limitations in further detail below.

- a. *“virtual computer systems, each of said virtual computer systems capable of accessing a selection of the partitions”*

Petitioner relies on Hermann to teach the recited “virtual computer systems . . . capable of accessing a selection of the partitions.” Pet. 46–47. Petitioner argues Hermann's choice of a hibernation image to restore is a selection of partitions because each hibernation image is associated with a particular operating system (and thus, is associated with certain partitions) in the hard disk. Pet. 46; Ex. 1002 ¶¶ 36–38. For example, Petitioner argues, if a user selects an OS/2 hibernation image, the user has selected partitions associated with the OS/2 operating system that the OS/2 operating system when active is capable of accessing. *Id.* Similarly, Petitioner argues, if the

user selects a Linux hibernation image, the user has selected a different set of partitions associated with and accessible by the active Linux OS. *Id.*

Patent Owner asserts that a “virtual computer system” has a unique master boot record (“MBR”) and its own partition table. PO Resp. 23. Patent Owner argues that this configuration provides protection between operating system environments, preventing one operating system environment from damaging another environment. *Id.* Patent Owner argues that “Hermann does not mention that each operating system includes a unique MBR and its own partition table.” *Id.*

Patent Owner’s argument is not persuasive because it is based on the district court’s claim construction, which we declined to adopt for the reasons discussed above. Under our construction, “virtual computer system” means “a set of partitions and file systems that are visible and available to a given operating system environment when that environment is actively executing on the physical computer.” In Hermann, each hibernation image or file is associated with a particular operating system environment and its related partitions. Ex. 1002 ¶¶ 70–71. The hibernation image or file in Hermann contains all the information necessary to restore a particular operating system environment. *Id.* ¶ 70 (citing Ex. 1005, 9:30–37). This includes an indication of which hard disk partitions are accessible to the OS environment and how those partitions are mapped and organized in the OS environment. *Id.* When leaving a hibernation state, e.g. returning to an active, or normal, operating state, a user has the option of choosing to resume one of several OS instances from corresponding hibernation images or files that were stored. *Id.* ¶ 51.

Accordingly, we are persuaded Herman teaches a set of partitions or files that are visible and available to a given operating system environment when that environment is actively executing on a physical computer. Ex. 1002 ¶ 96. Patent Owner’s argument that Hermann does not mention that each operating system includes a unique master boot record and its own partition table is not commensurate with the scope of the claimed limitation.

b. “virtual table of contents (VTOC) being capable of dynamically configuring a plurality of partition tables”

Petitioner relies on Hermann to teach the recited “virtual table of contents (VTOC) being capable of dynamically configuring a plurality of partition tables.” Pet. 47–48. Petitioner argues that Hermann teaches a VTOC in the form of hibernation images and corresponding control information depicted in Figure 6. *Id.* at 48; Ex. 1005, Fig. 6. Petitioner argues that in a multi-OS system, such as Hermann, the system must dynamically configure the partition tables (through use of a flag) to mark the primary partition of the current running operating system “active,” because, in general, only one partition in the system is marked “active” at a time. Pet. 47; *see* Ex. 1005, 7:49–54.

Petitioner argues that the information on which partition to mark active comes from the hibernation files shown in Figure 6, i.e. part of the virtual table of contents. Pet. 47; Ex. 1005, Fig. 6. Each hibernation file, Petitioner argues, is associated with a particular operating system. *Id.*; Ex. 1005, 11:38–50 (referring to an “OS/2 hibernation image” and a “Linux hibernation image”); Ex. 1002 ¶¶ 98, 101. Petitioner argues the operating system associated with the hibernation file will determine which partition is marked active. Ex. 1002 ¶¶ 71, 101. Therefore, Petitioner argues, the

partition data in Figure 6 of Hermann determines the dynamic configuration of partition tables. Pet. 48.

Patent Owner asserts that Hermann fails to describe a VTOC as defined in the '677 patent. PO Resp. 19. Patent Owner argues that the portion of Hermann relied upon by Petitioner illustrates information stored during creation of a hibernate image, and that “a hibernate image is *not* part of a VTOC, and is not ‘capable of dynamically configuring a plurality of partition tables.’” *Id.* Patent Owner argues that “the information saved by Hermann for hibernation is a recordation of state related to files and memory of a hibernated OS,” and that none of this information “is a specific data structure of a hard disk,” nor is it “defined by a Cabinet of the OS.” *Id.* at 21. Patent Owner argues that none of this information “is capable of *dynamically* configuring a plurality of partition tables,” noting that Petitioner allegedly admits that Hermann does not disclose using multiple partition tables. *Id.* Patent Owner argues that none of the information “can be used to rewrite a partition table, nor does Hermann discuss any such partition table rewriting functionality to switch between [operating systems].” *Id.*

Under our construction, “virtual table of contents” or “VTOC” means a “data structure maintained on secondary storage, but outside the primary file systems, that contains all necessary information about each virtual computer system environment of a physical computer, including storage information, partitions and file systems.” Pursuant to the claim language, the VTOC must be “capable of dynamically configuring a plurality of partition tables.”

Hermann teaches a VTOC in the form of hibernation images and corresponding control information depicted in Figure 6. Ex. 1005, Fig. 6.

The hibernation images and control information constitute an organized data structure, and within each hibernation image the data is organized into data blocks consisting of file allocation information, work data required for hibernation, content of volatile memory, and the content of main memory. Ex. 1005, 16:43–17:11; Ex. 1002 ¶ 98. The data is maintained in secondary storage (hard disk storage) and outside the primary file system. Ex. 1005, 16:34–37, 16:56–17:4; Ex. 1002 ¶¶ 96–97. The hibernation images contain all the necessary state information required to reestablish a respective OS instance and its corresponding processor states. Ex. 1005, 3:28–31, 9:30–37; Ex. 1002 ¶ 97.

In Hermann, the system dynamically configures partition tables (through use of a flag) to mark the primary partition of the current running operating system as “active,” because, in general, only one partition in the system is marked “active” at a time. Ex. 1005, 7:49–54. For example, if a user selects an OS/2 hibernation image to restore, a partition table with the partition containing the OS/2 operating system marked active will be configured. Ex. 1002 ¶¶ 72–74, 101. If the user then selects a Linux hibernation image to restore, a partition table with the partition containing the Linux operating system marked active will be configured. *Id.* Therefore, a plurality of partition tables are configured, i.e., one for the OS/2 image and one for the Linux image. These partition tables are configured dynamically depending on the user’s selection of hibernation images. Ex. 1002 ¶ 101.

Accordingly, we are persuaded Hermann teaches a “data structure maintained on secondary storage, but outside the primary file systems, that contains all necessary information about each virtual computer system

environment of a physical computer, including storage information, partitions and file systems, capable of dynamically configuring a plurality of partition tables.”

c. *“means for selecting one of said virtual computer systems to become next operable before suspending a currently operational virtual computer system”*

Petitioner relies on Hermann and the BootMagic User Guide to teach the recited “means for selecting one of said virtual computer systems to become next operable before suspending a currently operational virtual computer system.” Pet. 49–51. Petitioner argues that Hermann teaches offering a user a selection of available hibernation files and restoring the OS instance from that hibernation file. *Id.* at 49; Ex. 1005, 12:35–41. Petitioner argues the sequence of events is illustrated in Figure 7 of Hermann, where OS1 is running in box 80, wherein the user can opt to hibernate the current OS at box 81. *Id.* at 49, Ex. 1005, Fig. 7. Petitioner argues that if the user selects to hibernate the current operating system, the user has the option to perform a regular hibernation or switch to a different operating system. Pet. 49–50. The user has a selection of hibernation images from which to choose. Ex. 1005, 12:35–41; Pet. 50. After selecting a hibernation file, the current operating system is suspended and the selected hibernation file is restored and becomes operable. Ex. 1005, 12:35–44; Pet. 50.

Petitioner admits Hermann does not explicitly disclose the method of presenting the selection of hibernation files to the user and the method of user selection. Pet. 50; Ex. 1005, 12:35–41. For this aspect of the limitation, Petitioner relies on the BootMagic User Guide, which shows a graphical menu listing of the operating systems available on the system for selection. *Id.* at 50–51; Ex. 1007, xiv. Petitioner argues a person of ordinary

skill in the art would have been motivated to rely on a known user interface, such as BootMagic, which presents a selection of operating systems to a user and allows the user to choose among them. *Id.* at 50. Petitioner argues this would have been a design choice, and would thus have been an obvious feature to incorporate with Hermann with an expectation of success. *Id.* at 51; Ex. 1002 ¶¶ 75–82, 105.

Patent Owner asserts that Hermann fails to disclose any specifics of presenting a selection of hibernation files to a user. PO Resp. 25–26. Patent Owner also asserts that there is no mention of a GUI in Hermann, and therefore, no reference in Hermann leaving implementation details of a GUI to one of ordinary skill in the art. *Id.* at 26. Thus, Patent Owner argues, it cannot be a design choice to modify Hermann to include a GUI, because undue experimentation would be required. *Id.* According to Patent Owner, Hermann provides “no disclosure of how one would actually present and select hibernate files.” *Id.* at 27. No mention is made in Hermann, Patent Owner argues, of whether an interface would be graphical, what the options might be, or how a selection would be made. *Id.*

Patent Owner also argues Petitioner’s asserted combination of Hermann and the BootMagic Guide does not cure these deficiencies. *Id.* at 27–29. Patent Owner asserts that the BootMagic Guide “presents a menu at boot of unstarted operating systems to be booted,” and “[n]o operating system is currently operational when the BootMagic GUI is presented, and none would be suspended for a next operating system to become operable.” *Id.* at 28–29. Therefore, Patent Owner argues, no GUI is disclosed or suggested in the BootMagic Guide for selecting the next operable virtual computer system before suspending a currently operational system. *Id.*

Patent Owner's arguments are not persuasive because nonobviousness cannot be established by attacking references individually where, as here, the ground of unpatentability is based upon the teachings of *a combination* of references. *In re Keller*, 642 F.2d 413, 426 (CCPA 1981). Rather, the test for obviousness is whether the combination of references, taken as a whole, would have suggested the patentee's invention to a person of ordinary skill in the art. *In re Merck & Co., Inc.*, 800 F.2d 1091, 1097 (Fed. Cir. 1986).

Under our construction, a “means for selecting one of said virtual computer systems to become next operable before suspending a currently operational virtual computer system” is a “GUI, Super OS GUI, or equivalents,” that allow for “selecting one of said virtual computer systems to become next operable before suspending a currently operational virtual computer system.”

Hermann teaches offering a user a selection of available hibernation files and restoring the OS instance from that hibernation file. Ex. 1005, 11:29–12:41; Ex. 1002 ¶¶ 52–57, 103. The sequence of events is illustrated in Figure 7 of Hermann (shown above). If the user selects to hibernate the current operating system, the user has the option to perform a regular hibernation or switch to a different operating system. Ex. 1005, Fig. 7. The user has a selection of hibernation images from which to choose. *Id.* at 12:35–41. Resuming a hibernation file makes a virtual computer system operable because it will make a set of partitions and file systems available and accessible. Ex. 1002 ¶ 103. Similarly, suspending the currently active operating system will make a set of partitions and file systems that are current accessible, inaccessible. *Id.*

The BootMagic User Guide shows a graphical menu listing of the operating systems available on the system for selection. Ex. 1007, xiv. A person of ordinary skill in the art would have been motivated to rely on a known user interface, such as BootMagic, which presents a selection of operating systems to a user and allows the user to choose among them. Ex. 1002 ¶¶ 78, 80, 104. This would have been a design choice and would thus have been an obvious feature to incorporate with Hermann with a reasonable expectation of success. *Id.* ¶¶ 75–82, 105.

It was generally known at the time of the claimed invention that graphical interfaces were user-friendly and easy to use in consumer software products, and that graphical interfaces allowed the system to communicate information via graphical elements. *Id.* ¶ 80. The '677 patent acknowledges that using graphical user interfaces for managing multiple operating systems on a single computer was not new in the art. Ex. 1001, 2:15–19.

Petitioner's Declarant submits that a person of ordinary skill in the art would have had a reasonable expectation of success in using a graphical user interface such as the one described in the BootMagic User Guide with Hermann. Ex. 1002 ¶ 81. The BootMagic GUI is part of a custom boot loader, and a similar GUI would have been trivial to incorporate into the custom boot loader of Hermann to provide the same capability for selection of an operating system. *Id.* Programming a graphical interface with a menu and a selection of different options like the GUI in the BootMagic User Guide to use it with Hermann's system would have been a common programming exercise that a person of ordinary skill easily could have accomplished. *Id.* ¶ 82.

Accordingly, we are persuaded the combination of Hermann and the BootMagic User Guide teaches or suggests selecting a virtual computer system to become the next operable system before suspending the currently operational virtual computer system through use of a GUI.

d. “means for switching of the virtual computer systems using a switch flag and BIOS ACPI solutions, and without initialization of power-on self test (POST) in the BIOS”

Petitioner relies on Hermann and the ACPI Specification to teach the recited “means for switching of the virtual computer systems using a switch flag and BIOS ACPI solutions, and without initialization of power-on self test (POST) in the BIOS.” Pet. 53–56. Petitioner asserts that in Hermann, when hibernating an operating system, the user has the option to perform a regular hibernation or to switch to a different OS. *Id.* at 53; Ex. 1005, 12:30–44, Fig. 7. Petitioner asserts the user’s preference is stored in and used to set a switch flag in the form of an NVRAM variable. *Id.* at 54; Ex. 1008, 2.

Petitioner asserts that when switching operating systems, the user can select from hibernation files. *Id.*; Ex. 1005, 12:36–39. The selected hibernation image, Petitioner asserts, represents the virtual computer system to be loaded next, and is noted by the system. *Id.*; Ex. 1005, 12:55–56, 14:13–16; Ex. 1002 ¶ 112.

Petitioner asserts Hermann teaches the system then executes the suspend routine to suspend the currently active operating system and saves the operating system environment into a hibernation image. *Id.*; Ex. 1005, Fig. 7, 12:35–39; Ex. 1002 ¶ 113. The hibernation state, Petitioner argues, is a special variant of the suspend state. Pet. 54; Ex. 1005, 1:51–54.

Petitioner asserts Hermann teaches that switching operating systems occurs “without having to shut-down and re-boot the system” and that the “power-down cycle is not invoked.” *Id.*; Ex. 1005, 12:39–44, 16:31–34. Petitioner argues that a person of ordinary skill in the art would understand that there is no need to initialize the power-on self test (POST) in BIOS if the system was never powered down. *Id.* at 54–55; Ex. 1002 ¶¶ 58–59, 114. Petitioner argues Hermann’s teaching of switching without shut-down and re-boot constitutes a disclosure of proceeding without initializing the power-on self test (POST) in the BIOS. *Id.*

Petitioner asserts Hermann teaches that in the typical multi-OS situation, the partition containing the selected operating system is marked “active,” after which the normal boot process continues. Pet. 55; Ex. 1005, 7:49–54; Ex. 1002 ¶ 115. Petitioner argues marking a partition as “active” involves changing or updating the partition table, and because the partition table is a part of the master boot record (MBR), to mark a partition “active” is to adjust the MBR. *Id.*; Ex. 1002 ¶¶ 72–74, 115.

Petitioner asserts Hermann teaches invoking the BIOS resume procedure to restore the state of a previously suspended OS environment. Hermann teaches that “the dehibernation sequence restores the state of OS2 (box 83).” Pet. 56; Ex. 1005, 12:39–41. Hermann teaches that the dehibernation sequence involves using the APM BIOS Resume Routine and/or the Restore CPU State Routine. Ex. 1005, 9:13–25, Fig. 2; Ex. 1002 ¶ 116; Pet. 56. Therefore, Petitioner argues, Hermann teaches invoking the BIOS resume procedure to restore the state of a previously suspended OS environment, i.e. the OS2 hibernation file. Pet. 56.

Although the BIOS routines invoked by Hermann are Advanced Power Management (APM) routines and not ACPI BIOS routines, Petitioner argues it would have been obvious to a person of ordinary skill in the art to implement Hermann using ACPI BIOS routines instead of APM BIOS routines. *Id.*; Ex. 1014, 2:53–55. Combining Hermann with the ACPI Specification, Petitioner argues, teaches the use of ACPI BIOS routines to perform the hibernation/dehibernation routines disclosed therein. *Id.*; Ex. 1002 ¶¶ 83–86, 117.

Patent Owner contends that it would not have been obvious to one of ordinary skill in the art to implement the system of Hermann using the ACPI Specification in lieu of the APM (Advanced Power Management) approach mentioned in Hermann. PO Resp. 30–33. Patent Owner asserts that ACPI was designed to coexist with APM and that one of ordinary skill in the art would, therefore, not have been motivated to replace APM with ACPI. *Id.* at 31. Patent Owner argues that it took many years for some operating systems, such as Linux, to support ACPI, even though ACPI was initially released prior to the filing of Hermann. *Id.* at 31–32. Hermann’s reliance on APM even though ACPI had been previously released, Patent Owner argues, is evidence that one of ordinary skill in the art would not have substituted ACPI for APM with Hermann. *Id.* at 32.

Under our construction, the recited “means for switching of the virtual computer systems using a switch flag and BIOS ACPI solutions, and without initialization of power-on self test (POST) in the BIOS” is a processor performing an algorithm that 1) notes the virtual computer system to be loaded next and set the switch flag; 2) suspends the currently active OS environment; 3) proceeds without initialization of power-on self test (POST)

in the BIOS; 4) adjusts MBR to reflect the newly activated virtual computer system; and 5) invokes the BIOS resume procedure to restore the state of a previously suspended OS environment of the newly activated virtual computer system, and equivalents thereof.

As discussed further below, Petitioner shows that Hermann as combined with the ACPI Specification teaches the above-listed algorithmic structure. The ACPI Specification was publicly available in 1999 and describes its purpose as “an orderly transition from existing (legacy) hardware to ACPI hardware,” naming “APM” as one of its predecessors. Ex. 1006, 13. The ACPI Specification is backwards compatible and encompasses all that APM could do and more. *Id.*; Ex. 2018, 229:17–23. These developments provided one of ordinary skill in the art with the motivation to replace APM in existing applications with ACPI, which provides “enhance[ed] power management functionality and robustness.” *Id.* at 13. One of ordinary skill in the art, therefore, would have been motivated to use ACPI with Hermann’s hibernation/dehibernation routines because of its enhanced abilities.

Patent Owner’s argument that Linux may not have supported ACPI for a time does not negate the motivation of one of ordinary skill in the art to use ACPI to implement Hermann’s functionality instead of APM. Obviousness is an inquiry made in view of what a person of skill in the art understands and would be motivated to do at the time of the alleged invention, and a person of ordinary skill in the art would have understood both the interchangeability of APM and ACPI as well as the benefits of using ACPI instead of APM. *See, KSR Int’l. Co. v. Teleflex Inc.*, 550 U.S. 398, 421 (2007) (“When there is a design need or market pressure to solve a

problem and there are a finite number of identified, predictable solutions, a person of ordinary skill has good reason to pursue the known options within his or her technical grasp.”).

Petitioner provides persuasive technical evidence and argument that the combination of Hermann and the ACPI Specification provides the modified BIOS code and custom boot loader operations necessary to implement the algorithm described above. See Ex. 1002 ¶¶ 36–38, 55, 58–59, 66–88, 96, 112–118; Ex. 1105, 1:51–54, 3:32–49, 9:13–16, 35–58, 1057–11:8, 11:29–50, 12:22–44.

Accordingly, we are persuaded that one of ordinary skill in the art would have been motivated to combine Hermann with the ACPI Specification, and that this combination teaches or suggest the modified BIOS code and custom boot loader operations that provide a “means for switching of the virtual computer systems using a switch flag and BIOS ACPI solutions, and without initialization of power-on self test (POST) in the BIOS”.

e. “wherein the switch flag is a flag that is set up in storage to differentiate between suspend for fast switching and power save suspend”

Petitioner relies on Hermann to teach the recited “wherein the switch flag is a flag that is set up in storage to differentiate between suspend for fast switching and power save suspend.” Pet. 57–59. Hermann teaches providing the user the option of performing a regular hibernation or switching to a different operating system. Ex. 1005, 12:30–44, Fig. 7 (box 81 depicting a user given a choice between inventive hibernate/dehibernate, box 82, and regular hibernate/dehibernate, box 84). Petitioner argues a person of skill in the art would understand that one way to store the user’s

choice between the inventive hibernate/dehibernate and regular hibernate/dehibernate is the use of a “flag”—a variable to mark a particular condition or status. Pet. 57; Ex. 1002 ¶¶ 68, 118.

Petitioner argues Hermann teaches a similar flag that those of skill in the art would understand to serve this purpose. *Id.* Petitioner argues that a person of skill in the art would understand from reading Hermann that an NVRAM variable (described in an IBM Technical Disclosure Bulletin (Ex. 1008) incorporated by Hermann) would store information about the type of hibernation that was requested. *Id.* at 57–58; Ex. 1002 ¶¶ 66–67. Petitioner argues a person of ordinary skill in the art would consider the NVRAM variable used in this manner as a “flag.” *Id.* at 58; Ex. 1002 ¶ 68. Petitioner argues the “regular hibernate/dehibernate” sequence shown in box 84 of Figure 7 is a “power save suspend” and that the “inventive hibernate/dehibernate” sequence shown in box 82 of Figure 7 is a suspend for “fast switching.” *Id.*; Ex. 1005, 1:39–2:6.

Patent Owner asserts that “[t]here is no mention of an NVRAM variable in Hermann.” PO Resp. 34. Patent Owner argues that the NVRAM variable mentioned in the IBM Technical Bulletin, which is incorporated by reference in Hermann, indicates only whether or not there was a previous hibernate. *Id.* at 34–35. Patent Owner argues the IBM Technical Bulletin does not contemplate fast switching, and therefore, does not disclose that a user’s preference to perform regular hibernation or to switch to a different operating system is stored in the NVRAM variable. *Id.* at 35. Thus, Patent Owner argues, Hermann fails to teach or suggest the limitation “wherein the switch flag is a flag that is set up in storage to differentiate between suspend for fast switching and power save suspend.” *Id.* at 36.

The IBM Technical Disclosure Bulletin (Ex. 1008), which Hermann incorporates by reference (Ex. 1005, 15:17–22), teaches the use of NVRAM variables in systems prior to Hermann’s invention that “perform[ed] a complete system hibernation and resume that can be independent of the operating system and hardware resident firmware.” Ex. 1008, 143. The IBM Bulletin notes that the “act of resuming from hibernation is also simplified” because the system “detects from a previously saved NVRAM variable that it previously hibernated.” *Id.* at 144. Hermann incorporates the IBM Bulletin’s teaching as computer architecture support for its inventive switching concepts using NVRAM variables to store the information needed to manage hibernation. Ex. 1005, 15:17–22; Ex. 1002 ¶¶ 65–70, 112.

Used in this manner, the NVRAM variable operates as a switch flag because it distinguishes between a state in which the system switches operating systems and a state in which the system does not switch operating systems. *Id.* ¶ 118.

Accordingly, we are persuaded the combination of references teaches or suggests a switch flag that is set up in storage to differentiate between suspend for fast switching and power save suspend.

f. Summary as to disputed limitations of Claim 1

For the reasons stated above, we are persuaded that the combination of Hermann, the BootMagic User Guide, and the ACPI Specification teaches or suggests the following limitations as recited in claim 1: “a plurality of virtual computer systems, each of said virtual computer systems capable of accessing a selection of the partitions,” a “virtual table of contents (VTOC) being capable of dynamically configuring a plurality of partition tables,” “means for selecting one of said virtual computer systems to become next

operable before suspending a currently operational virtual computer system,” “means for switching of the virtual computer systems using a switch flag and BIOS ACPI solutions, and without initialization of power-on self test (POST) in the BIOS,” and “wherein the switch flag is a flag that is set up in storage to differentiate between suspend for fast switching and power save suspend.”

2. Independent Claim 6

Claim 6 recites a method of managing a computer system and fast switching between operating systems, the computer system having a plurality of operating systems and a plurality of virtual computer systems and at least one cabinet for isolating each virtual computer system from other virtual computer systems, each of said operating systems configured on a corresponding virtual computer system. Ex. 1001, 16:65–18:5. Claim 6 recites limitations similar to those of claim 1, but in a method format having a series of steps such as a “fast suspending step,” and an “activating step.” *Id.*

With respect to claim 6, Petitioner relies on the same teachings of Hermann, the BootMagic User Guide, and the ACPI Specification discussed above with respect to claim 1 to demonstrate that the recited limitations of claim 6 would have been obvious to one of ordinary skill in the art. Pet. 70–73.

Patent Owner contends that similar arguments it made with respect to claim 1 apply to the patentability of claim 6. PO Resp. 14. Patent Owner’s arguments are not persuasive for the reasons discussed above. We conclude that Petitioner has established by a preponderance of the evidence that claim

6 would have been obvious over the combination of Hermann, the BootMagic User Guide, and the ACPI Specification.

3. Dependent Claims 2 and 7

Claim 2 depends from claim 1. Claim 7 depends from claim 6. Ex. 1001, 16:30–18:10. These dependent claims recite “fast resuming” limitations. *Id.* Claim 2 recites “[t]he hardware platform as in claim 1, wherein: said means for making the selected virtual computer system operable comprising means for fast resuming said selected virtual computer system.” Claim 7 recites “[t]he method as in claim 6, wherein said activating step further comprises: fast resuming said another virtual computer system to a current state.”

Petitioner explains how Hermann teaches the recited limitations, and relies on some of the same evidence and arguments made with respect to independent claims 1 and 6. *See* Pet. 60, 73. Petitioner shows persuasively that the combination of Hermann, the BootMagic User Guide, and the ACPI Specification would have rendered claims 2 and 7 obvious. *See id.* Patent Owner does not contest separately the patentability of claims 2 and 7.³ Although the burden still remains with the Petitioner to establish these claims are unpatentable by a preponderance of the evidence, Patent Owner has waived its arguments with respect to the patentability of claims 2 and 7.

We construed “fast resume” to mean “making modified use of the BIOS power management standard to allow transfer from suspended state to

³ The only reference to claims 2 or 7 that we can find in Patent Owner’s Response is the title to Section VI, which reads “Hermann, The BootMagic User Guide, and the ACPI Specification do not Render Claims 1-2 and 6-7 of the ’677 Patent Obvious.” *See* PO Resp. 14, 27 (emphasis omitted). We find no reference to “fast resuming” in Patent Owner’s Response.

operating state without POST and without loading all components of an operating system from the cold boot state.” Hermann teaches dehibernating a hibernation file selected by a user. Ex. 1005, 11:45–50. The dehibernation process activates the OS instance stored within the hibernation file and activates the set of partitions and files systems associated with that OS to restore it to a current active state. Ex. 1002 ¶¶ 120, 148.

We are persuaded, by a preponderance of the evidence, that the combination of references teaches the recited “fast resuming” limitations.

4. Indicia of Nonobviousness

Patent Owner provides a series of observations concerning technologies used by other entities under a section titled “Secondary Indications of Non-Obviousness.” PO Resp. 36–48. These observations include a description of online services offered by Amazon Web Services AWS (*id.* at 40–41), eBay Open Stack (*id.* at 42–43), and Alibaba Cloud (*id.* at 43–44).

Patent Owner argues a nexus between these services and the claimed invention by alleging, “Amazon has had so much success with this platform that they have created their cloud services through Amazon Web Services that provide their customers with the ability to access their clouds and store in their cloud services through the use of the claim limitations of the ’677 [p]atent.” PO Resp. 45–46. Patent Owner also alleges that AWS admits practicing the claimed invention because it advertises that “[t]he AWS Cloud provides a broad set of infrastructure services, such as computing power, storage options, networking and databases, delivered as a utility: on-demand, available in seconds, with pay-as-you-go pricing.” *Id.* at 46 (quoting Ex. 2025, <https://aws.amazon.com/what-is-aws/>). Aside from

attorney argument, Patent Owner provides no evidence that these commercial systems utilize the claimed invention. *See id.* at 44–47; Pet. Reply 20–22. Patent Owner, for example, does not map any of the limitations recited in claim 1 to any particular feature of these commercial systems. *See id.*

Assuming, *arguendo*, that these services use the claimed invention, Patent Owner has not presented any evidence that the claimed invention contributes to the commercial success of these systems. Patent Owner argues that “the fast switching afforded by the [']677 [p]atent has provided each of these companies with the ability to provide their customers with an incredible amount of data at the tip of their fingertips at amazing speeds.” *Id.* at 48. The quote from AWS’s website, however, indicates that its cloud service constitutes “a broad set of infrastructure services” and other items. Ex. 2025. The allegedly “fast switching afforded by the ’677 patent,” even if true, therefore, would be only one feature of a much larger system. Patent Owner does not show these services employed the same switching technique, and does not quantify any switching speeds. Even assuming, however, some switching mechanism is employed by these services, “[w]hen the thing that is commercially successful is not coextensive with the patented invention—for example, if the patented invention is only a component of a commercially successful machine or process—the patentee must show prima facie a legally sufficient relationship between that which is patented and that which is sold.” *Demaco Corp. v. F. Von Langsdorff Licensing Ltd.*, 851 F.2d 1387, 1392 (Fed. Cir. 1988); *see also Polaris Industries, Inc. v. Artic Cat, Inc.*, 882 F.3d 1056 (Fed. Cir. 2018) (citing *Demaco*, 851 F.2d at 1392, and stating “[p]ut another way, ‘objective

evidence of non-obviousness must be commensurate in scope with the claims which the evidence is offered to support.” (quoting *In re Grasselli*, 713 F.2d 731, 743 (Fed. Cir. 1983)).

In a similar fashion, Patent Owner provides brief descriptions of alleged commercial success and unresolved need concerning Flash Vos, Inc. (“Flash Vos, Inc. moved the computer industry a quantum leap forward in the late 90’s when it invented Systems Virtualization”) and GEMSA (“GEMSA creative achievements not only revolutionized the development of virtualization technology for support of multiple operating systems but also helped the development of internet advertising and information accessing from multiple data sources”). PO Resp. 47–48.

These attorney arguments, however, are not supported by any evidence of commercial success. Patent Owner has not provided, for example, any evidence that Flash Vos or GEMSA generated significant sales or was otherwise commercially successful. Moreover, Patent Owner has not persuasively established a nexus between the challenged claims and Flash Vos’ or GEMSA’s alleged commercial success.

Even assuming Flash Vos or GEMSA products or services embodied the claimed invention and presuming that any commercial success of Flash Vos or GEMSA was due to the claimed subject matter, Patent Owner’s argument fails because Patent Owner has put forth no evidence that Flash Vos or GEMSA actually had any commercial success.⁴ The record contains

⁴ See *J.T. Eaton & Co. v. Atl. Paste & Glue Co.*, 106 F.3d 1563, 1571 (Fed. Cir. 1997) (presuming, based on a requisite showing of a nexus between the patented invention and commercial success, that commercial success of patentee’s products was due to the patented invention).

no evidence of sales, revenue, profits, or any other indicia of commercial success of Flash Vos, GEMSA, or its products or services.

Patent Owner also alleges copying of the claimed invention by asserting that “[a]t least Amazon, eBay and Alibaba have utilized the claimed inventions of the [’]677 [p]atent.” PO Resp. 48. According to Patent Owner, “the fast switching afforded by the ’677 [p]atent has provided each of these companies with the ability to provide their customers with an incredible amount of data at the tip of their fingertips at amazing speeds.” *Id.* These attorney arguments are not supported by evidence either. Patent Owner provides no evidence, for example, that any of Amazon, eBay, or Alibaba were even aware of the ’677 patent, Flash Vos, or GEMSA, much less copied the invention claimed therein.

Patent Owner’s unsupported assertions of commercial success, unresolved need, and copying, along with a lack of evidence or analysis establishing a nexus between the claims at issue and any product or service alleged to be practicing the claimed invention, fail to provide substantive evidence of nonobviousness. Petitioner’s obviousness showing considered in conjunction with no nexus, or, at most, a weak nexus, establishes by a preponderance of evidence that challenged claims would have been obvious. *See ClassCo, Inc. v. Apple, Inc.*, 838 F.3d 1214, 1221 (Fed. Cir. 2016) (“[T]he Board should have afforded ClassCo’s evidence some weight, taking into account the degree of the connection between the features presented in evidence and the elements recited in the claims. There is no hard-and-fast rule for this calculus, as ‘[q]uestions of nexus are highly fact-dependent and, as such are not resolvable by appellate-created categorical rules and hierarchies as to the relative weight or significance of proffered evidence.’”)

(quoting *WBIP, LLC v. Kohler Co.*, 829 F.3d 1317, 1331 (Fed. Cir. 2016)); *id.* at 1220 (“[T]here is no nexus unless the evidence presented is ‘reasonably commensurate with the scope of the claims.’”) (quoting *Rambus Inc. v. Rea*, 731 F.3d 1248, 1257 (Fed. Cir. 2013)).

III. CONCLUSION

Based on all the evidence of record, we determine that Petitioner has established by a preponderance of the evidence that claims 1, 2, 6, and 7 of the '677 patent would have been obvious under 35 U.S.C. § 103(a) over the teachings of Hermann, the BootMagic User Guide, and the ACPI Specification.

IV. ORDER

For the reasons given, it is

ORDERED that claims 1, 2, 6, and 7 of the '677 patent are unpatentable; and

FURTHER ORDERED that parties to the proceeding seeking judicial review of the Final Written Decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.

IPR2016-01829
Patent 7,356,677 B1

PETITIONER:

Benjamin Weed
Jackson Ho
K&L GATES LLP
benjamin.weed.ptab@klgates.com
jackson.ho@klgates.com

Todd Siegel
Andrew Mason
KLARQUIST SPARKMAN, LLP
todd.siegel@klarquist.com
andrew.mason@klarquist.com

Brett Watkins
Lance Yang
QUINN EMANUEL URQUHART & SULLIVAN, LLP
brettwatkins@quinnemanuel.com
lanceyang@quinnemanuel.com

PATENT OWNER:

Lissi Mojica
Mark Cantor
Sangeeta Shah
Brian C. Doughty
Richard J. Cantor
BROOKS KUSHMAN P.C.
lmojica@brookskushman.com
mcantor@brookskushman.com
sshah@brookskushman.com
bdoughty@brookskushman.com
rjcantor@BrooksKushman.com
GEMSA0102IPR1@brookskushman.com