

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

AMERICAN MEGATRENDS, INC., et al.,
Petitioner,

v.

KINGLITE HOLDINGS INC.,
Patent Owner.

Case IPR2015-01515
Patent 5,978,912

Before GLENN J. PERRY, TREVOR M. JEFFERSON, and
BRIAN J. McNAMARA, *Administrative Patent Judges*.

PERRY, *Administrative Patent Judge*.

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

I. INTRODUCTION

This is a Final Written Decision entered in an *inter partes* review instituted pursuant to 35 U.S.C. § 314. For reasons discussed below, we determine that Petitioner has shown by a preponderance of the evidence that some, but not all, of the claims at issue of U.S. Patent No. 5,978,912 (Ex. 1001, “the ’912 patent”) are unpatentable.

A. Procedural History

American Megatrends, Inc., Micro-Star International Co. Ltd, MSI Computer Corp., Giga-Byte Technology Co., Ltd., and G.B.T., Inc. (“Petitioner”) filed a Petition (Paper 7, “Pet.”) to institute *inter partes* review of claims 1–60 of the ’912 patent. 35 U.S.C. § 311. Patent Owner, Kinglite Holdings Inc., timely filed a Preliminary Response (Paper 12, “Prelim. Resp.”) contending that the Petition should be denied as to all. We instituted trial as to claims 1–10, 12–15, 17–19, 21–30, 32–45, and 47–60. Paper 16, “Dec. to Inst.” Patent Owner filed a Patent Owner Response (Paper 22, “PO Resp.”), and Petitioner filed a Petitioner Reply (Paper 23, “Pet. Reply”). Petitioner also requested rehearing of our Decision to Institute. Paper 18. In our rehearing decision, we added claims 11, 16, and 46 to the review. Paper 24, “Dec. on Reh’g.” We authorized and received from Patent Owner a Supplemental Response directed to the added claims (Paper 26, “Supp. Resp.”) and a Petitioner Reply to that Supplemental Response. Paper 27, “Reply to Supp. Resp.” We heard oral argument on October 17, 2016 and entered a transcript of that argument into the record. Paper 29 (“Tr.”).

B. Related Matters

American Megatrends reports the following related matters in civil litigation:

Kinglite Holdings Inc. v. Giga-Byte Tech. Co. Ltd., et al., CV 14-04989 JVS (PJWx) (C.D. Ca); and

Kinglite Holdings Inc. v. Micro-Star Int'l Co. Ltd., et al., CV 14-03009 JVS (PJWx) (C.D. Ca). Paper 6, 1.

American Megatrends reports the following related matters before the Board:

American Megatrends Inc., et al. v. Kinglite Holdings Inc., IPR2015-01079 (U.S. Pat. No. 6,373,498);

American Megatrends Inc., et al. v. Kinglite Holdings Inc., IPR2015-01081 (U.S. Pat. No. 5,987,604);

American Megatrends Inc., et al. v. Kinglite Holdings Inc., IPR2015-01094 (U.S. Pat. No. 6,401,202);

American Megatrends Inc., et al. v. Kinglite Holdings Inc., IPR2015-01132 (U.S. Pat. No. 6,523,123);

American Megatrends Inc., et al. v. Kinglite Holdings Inc., IPR2015-01133 (U.S. Pat. No. 5,732,268);

American Megatrends Inc., et al. v. Kinglite Holdings Inc., IPR2015-01140 (U.S. Pat. No. 6,519,659);

American Megatrends Inc., et al. v. Kinglite Holdings Inc., IPR2015-01141 (U.S. Pat. No. 6,633,976);

American Megatrends Inc., et al. v. Kinglite Holdings Inc., IPR2015-01188 (U.S. Pat. No. 5,836,013);

American Megatrends Inc., et al. v. Kinglite Holdings Inc., IPR2015-01189 (U.S. Pat. No. 5,836,013);

American Megatrends Inc., et al. v. Kinglite Holdings Inc., IPR2015-01191 (U.S. Pat. No. 6,892,304);

American Megatrends Inc., et al. v. Kinglite Holdings Inc., IPR2015-01197 (U.S. Pat. No. 6,487,656);

American Megatrends Inc., et al. v. Kinglite Holdings Inc., IPR2015-01487 (U.S. Pat. No. 6,308,265); and

American Megatrends Inc., et al. v. Kinglite Holdings Inc., IPR2015-01488 (U.S. Pat. No. 7,185,189). *Id.* at 1–2.

C. The '912 Patent (Ex. 1001)

1. Described Invention

The '912 patent describes a computer that, prior to booting its operating system or after an operating system failure, can communicate with a networked computer. Ex. 1001, Abstract. A “multitasking kernel” that supports communication is implemented in a network-enhanced BIOS. *Id.*

Figure 4 of the '912 patent is reproduced below.

Fig 4.

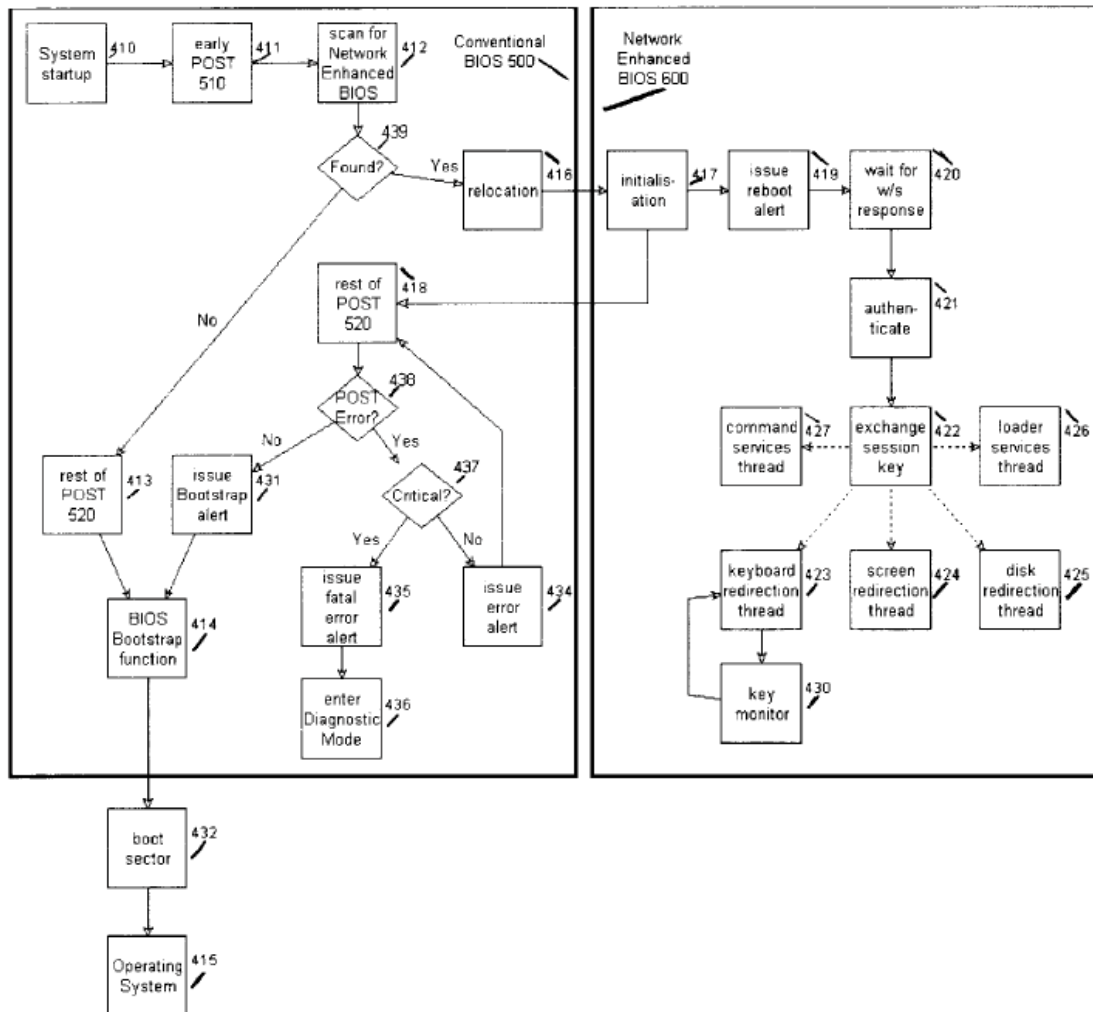


Figure 4 depicts a flowchart showing software contained on a computer configured for communication in accordance with a preferred embodiment.

Figure 4 explains operation of “network-enhanced BIOS” 600, which works with “conventional BIOS” 500 to provide network communication in the absence of a functional operating system. Ex. 1001, 7:1–50. At start-up 410, control passes to “Early Power On Self Test” (POST) 510 which performs preliminary tasks such as programming the memory controller,

interrupt controller, system bus controller, and other chipset registers. At 412 control passes to the network-enhanced BIOS installation check routine¹ which scans non-volatile memory 125 looking for a signature indicating the start of network-enhanced BIOS 600 code (right side Figure 4). If no signature is present, the rest of POST routine 520 executes at 413 and control passes at 414 to an operating system (OS) bootstrapping routine. At 432 the bootstrapping routine loads the OS and at 415 passes CPU control to the OS. Ex. 1001, 7:1–18.

If a network-enhanced BIOS 600 code signature is present, a relocation routine 416 copies the network-enhanced BIOS 600 from non-volatile memory 125 into RAM 120, and CPU control passes, at 417, to initialization routine 610 of the network-enhanced BIOS 600 code stored at an address at a known offset from the signature. Because conventional BIOS 500 typically operates in real mode, the network-enhanced BIOS initialization routine 610 switches the CPU into protected mode and initializes 32-bit kernel 620, and each component of protocol stack 640. *Id.*, 7:19–30.

Network-enhanced BIOS kernel 620 is preferably multithreaded. A task may be prevented from executing if waiting for an external event or another task on which it depends. To manage tasks, the kernel provides functions for signal, semaphore and mutual exclusion. Kernel dispatcher 621 switches CPU control among tasks. *Id.*, 7:1–18, 7:31–34.

¹ The check routine is part of enabler 560 shown in Figure 3 of the '912 patent.

Plural 32-bit protected mode threads, started by the network-enhanced BIOS initialization routine 610, support network protocol stack 640, and communication with management workstation 200 application. Ex. 1001, 7:44–50.

2. Illustrative Claim

Claims 1, 37, 38, 51, 52, 57, and 58 are independent. Claim 1 of the '912 patent is illustrative of the challenged claims:

1. A method of operating a computer, said computer including a central processing unit (CPU) and a network interface coupling said computer to a network, said method comprising:

 automatically executing an initial start-up software on said CPU upon CPU system start-up;

 loading and executing a network-enhanced software on said CPU on completion of at least a first portion of said start-up software, said network-enhanced software supporting a protocol for communicating with workstations coupled to said computer network via said network interface; and

 loading and executing an operating system software on said CPU after said loading of said network-enhanced software, said operating system software being accessible by user-level application software programs executing on said CPU.

D. Instituted Challenges

We instituted *inter partes* review, as to all claims, except claims 20 and 31, as set forth in the following table:

Reference(s)	Basis	Claim(s) challenged
Chang ² and Dent ³	35 U.S.C. § 103	1–18, 24–30, 32, 35–53, 55, 56, 59 and 60
Chang, Dent, and Bizzarri ⁴	35 U.S.C. § 103	19, 21–23, 28, 29 and 54
Chang, Dent, and Aegis ⁵	35 U.S.C. § 103	33 and 34
Flaherty ⁶ and Chang	35 U.S.C. § 103	57 and 58

II. DISCUSSION

A. Claim Interpretation

The Board interprets unexpired claims using the “broadest reasonable construction in light of the specification of the patent in which [they] appear[.]” 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 interpretation standard as the claim interpretation standard to be applied in *inter partes* reviews). Under this standard, we interpret claim

² U.S. Patent 5,444,850, issued August 22, 1995 (Ex. 1003, “Chang”).

³ U.S. Patent 5,884,073, issued March 16, 1999 (Ex. 1004, “Dent”).

⁴ U.S. Patent 5,732,268, issued March 24, 1998 (Ex. 1008, “Bizzarri”).

⁵ Arbaugh, William A; Farber, David J.; and Smith, Jonathan M., *A Secure and Reliable Bootstrap Architecture*, University of Pennsylvania Scholarly Commons, December 2, 1996 (Ex. 1014, “Aegis”).

⁶ U.S. Patent 5,146,568, issued September 8, 1992 (Ex. 1006, “Flaherty”).

terms using “the broadest reasonable meaning of the words in their ordinary usage as they would be understood by one of ordinary skill in the art, taking into account whatever enlightenment by way of definitions or otherwise that may be afforded by the written description contained in the applicant’s specification.” *In re Morris*, 127 F.3d 1048, 1054 (Fed. Cir. 1997). We generally presume that claim terms have their ordinary and customary meaning. *See Trivascular, Inc. v. Samuels*, 812 F.3d 1056, 1062 (Fed. Cir. 2016) (“Under a broadest reasonable interpretation, words of the claim must be given their plain meaning, unless such meaning is inconsistent with the specification and prosecution history.”); *In re Translogic Tech., Inc.*, 504 F.3d 1249, 1257 (Fed. Cir. 2007) (“The ordinary and customary meaning is the meaning that the term would have to a person of ordinary skill in the art in question.” (internal quotation marks omitted)). A patentee, however, may rebut this presumption by acting as his or her own lexicographer, providing a definition of the term in the specification with “reasonable clarity, deliberateness, and precision.” *In re Paulsen*, 30 F.3d 1475, 1480 (Fed. Cir. 1994).

In our Decision to Institute, we preliminary construed the following claim terms:

Claim Term	Preliminary Construction
“initial start-up software”	“a set of instructions that initializes a computer system”
“network-enhanced software (NES)”	“software that facilitates networking”
“operating system (OS) software”	“system-level software that controls the execution of user level programs and that provides services to such user-level programs such as resource allocation, scheduling, I/O control and data management”
“power on self-test (POST)”	“the testing and initialization phases of the BIOS that begins when a system powers on”

Patent Owner continues to argue for a more restrictive construction of “network-enhanced software” that would require various specific functions of software described in the Specification. PO Resp. 8. However, we do not read into this claim term limitations of particular embodiments in the absence of a reason to do so. Here there is none. The term “network-enhanced software” appears only in the claims and not elsewhere in the Specification. We find no description of the term “network-enhanced software” in the Specification that would rise to the level of a definition. Embodiments are described as including a “network-enhanced BIOS.” The term “BIOS” appears in some, but not all, of the claims. We regard the term “network-enhanced software” as being broader in scope than “network-enhanced BIOS” based on our understanding that the term BIOS connotes to computer engineers specialized software particular to the rudimentary operation of a computer. We therefore adopt our preliminary constructions

along with our reasoning expressed in our Decision to Institute for purposes of this decision. Dec. to Inst. 12–14.

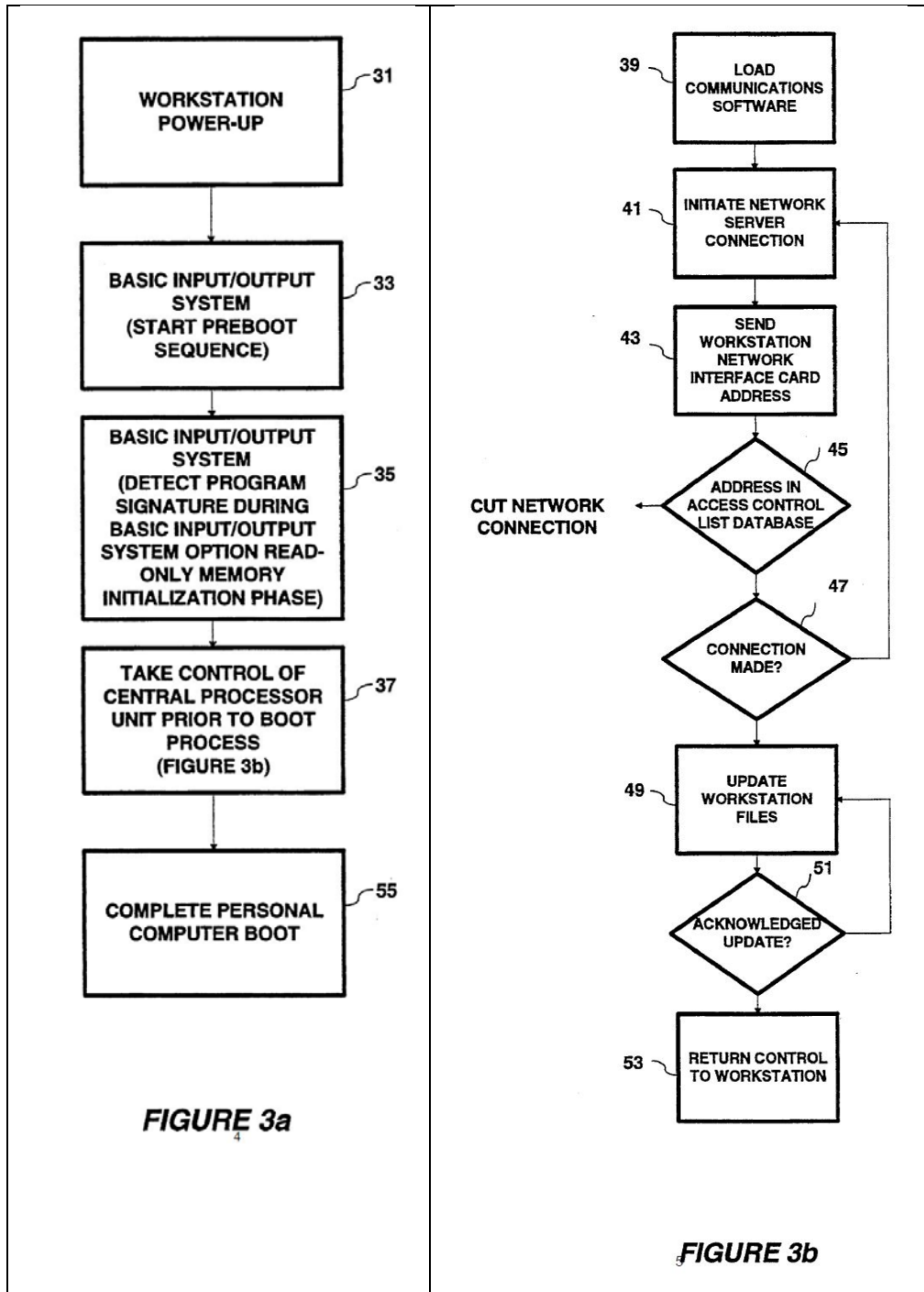
B. Overview of the References

We provide an overview of the references discussed in the specific challenges.

1. Chang

Chang describes information and file transfer between a computer, not yet booted to its local operating system, and a network computer. Ex. 1003, Abstract. To enable communication, a ROM or PROM containing code is placed in either 1) an unused boot ROM socket of a LAN card, or 2) a motherboard. *Id.*

During a boot sequence and prior to loading its operating system (OS), the computer performs certain functions using its BIOS to enable communication via the network. *Id.* This process is described at column 6, line 49 through column 7, line 36, with reference to Figures 3a and 3b, reproduced below.



Figures 3a and 3b depict flow charts showing processing carried out by Chang at system startup.

The boot process interrupts after the BIOS executes but before the Disk Operating System (DOS) executes. At power-up (step 31), the system BIOS is executed (step 33). The system BIOS detects the existence of a program in a ROM socket on a network interface card (step 35) and passes control to this program (step 37). Network communications software loads from the workstation storage medium (step 39) using its internal mini-operating system and the appropriate network communications protocols.

Communication with the Server Management Application (SMA) is initiated (step 41). Under instructions from the SMA, the invention executes the appropriate functions contained in the executive services engine of the program in the PROM. The executive services engine uses low-level control functions (e.g., BIOS under DOS) to perform the instructions sent by the server SMA to the workstation. After the server connects in step 41, the workstation NIC address is verified (step 43). If the NIC address is not in the address in access control list database (ACL-DB) (step 45), the connection to the network is cut, otherwise an attempt is made to connect the workstation to the network (step 47) and, if successful, workstation files are updated (step 49). Otherwise, the attempt to initiate the network connection is repeated (step 41). At step 49, the server sends any required workstation updates to the client and performs any assigned tasks under control from the SMA. During update, processes programmed in the PROM are executed and acknowledged (ACK) (step 51) by the firmware under instruction from the server SMA. The server and client communicate directly with workstation hardware and firmware at a level below the operating system, thus providing a sterile, controlled operating environment. Control then

returns to the workstation boot process (step 53) and the normal boot process is completed (step 55). Steps 39 and 41 are performed by file handler 27a and 27b. Steps 43, 45, 47, 49, 51, and 53 are performed by network services 27c and executable services engine 23. Ex. 1003, 6:49–7:36.

The firmware kernel provides the workstation with a pristine operating environment. The workstation can execute any instructions sent by the SMA, exit from control of the firmware, and commence normal boot. Changes made during the pre-boot process by the SMA are reflected immediately at the workstation.

2. Dent

Dent describes altering a boot sequence provided by a Basic Input/Output System (BIOS) to enable diagnostics of the electronic system containing the BIOS by a service provider. Access to the service provider is established through a publicly accessible network (e.g., Internet). Ex. 1004, Abstract. Dent's Figure 4 is reproduced below.

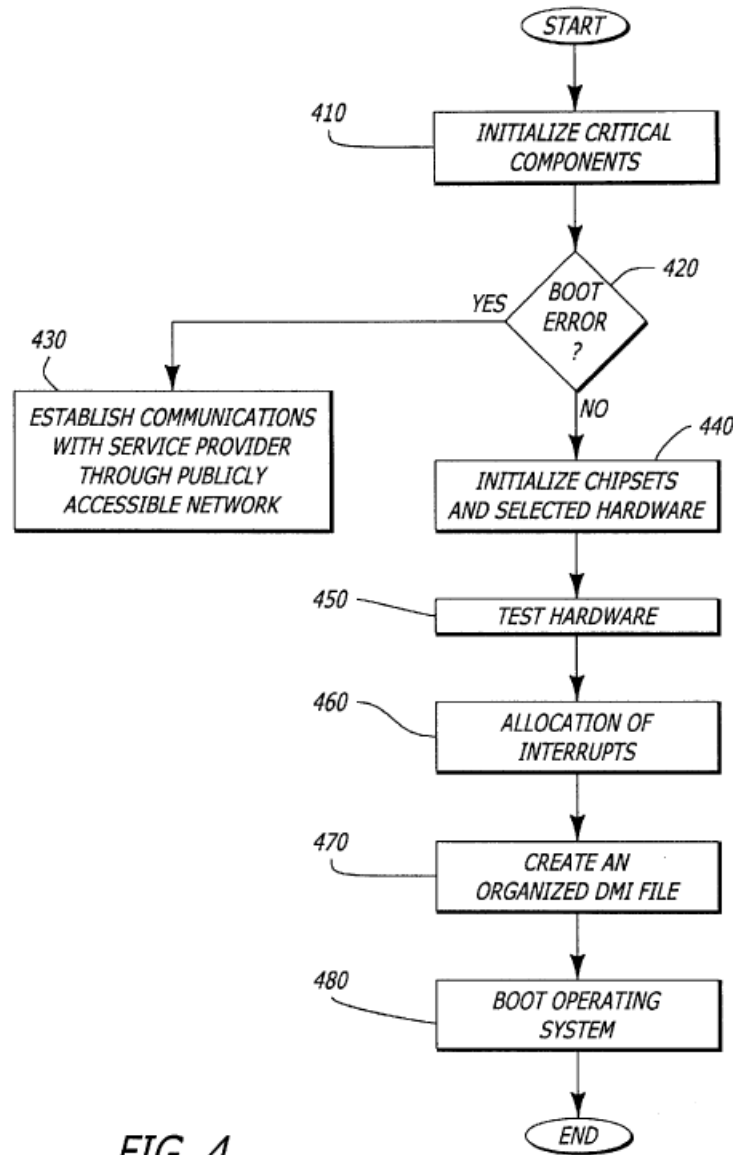


FIG. 4

Figure 4 depicts a flowchart illustrating a boot sequence that establishes an interconnection over a network for remote servicing of the electronic system upon experiencing a boot error.

As shown in Dent's Figure 4, after the electronic system has been powered, it undergoes a boot sequence in which components critical to establishing remote access through a publicly accessible network are

initialized before proceeding with the standard boot sequence (step 410). If it is necessary to service a boot error in the electronic system before proceeding with the standard boot sequence, remote communications are established over the Internet (steps 420 and 430). Otherwise, the boot sequence proceeds as discussed above by performing an initialization, testing and allocation phases before loading the operating system (steps 440–480). Ex. 1004, 5:1–15.

3. Bizzarri

Bizzarri describes an extended BIOS to establish remote communication for diagnostics and repair. Ex. 1008, Title.

Bizzarri's Figure 2 is reproduced below.

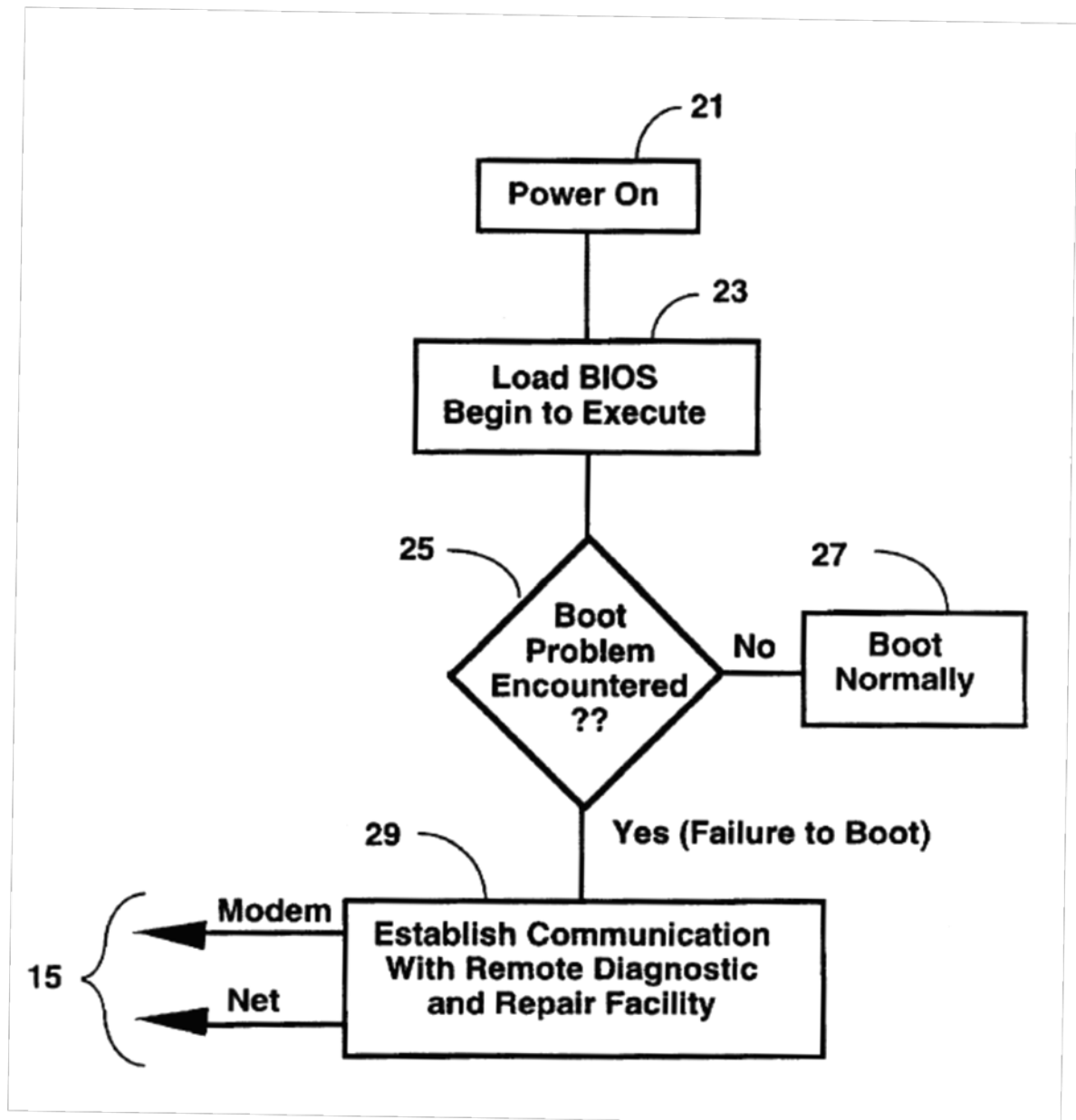


Figure 2 depicts a logic flow diagram of operation of an E-BIOS PC at boot up.

After power on at step 21, the E-BIOS is loaded into RAM and begins to execute at step 23. At step 25 E-BIOS execution continues, performing

POST and other BIOS startup functions and monitors for any fault that will prevent completion of the boot process. If no fault prevents normal system operation, boot continues. If a fault prevents normal system operation, boot proceeds at step 27 to presenting on a display monitor the interactive interface allowing a user to direct continuing operation. Control is diverted to step 29 and the E-BIOS establishes communication with a remote E-BIOS diagnostic and repair unit (item 13 in Figure 1). Ex. 1008, 5:66–6:26.

4. Aegis

Aegis is an academic article describing a secure and reliable bootstrap architecture. Ex. 1014, cover page. It describes the “AEGIS” architecture for initializing a computer system. It validates integrity at each layer transition in the bootstrap process. AEGIS also includes a recovery process for integrity check failures. Ex. 1014, Abstract.

5. Flaherty

Figure 1 of Flaherty is reproduced below.

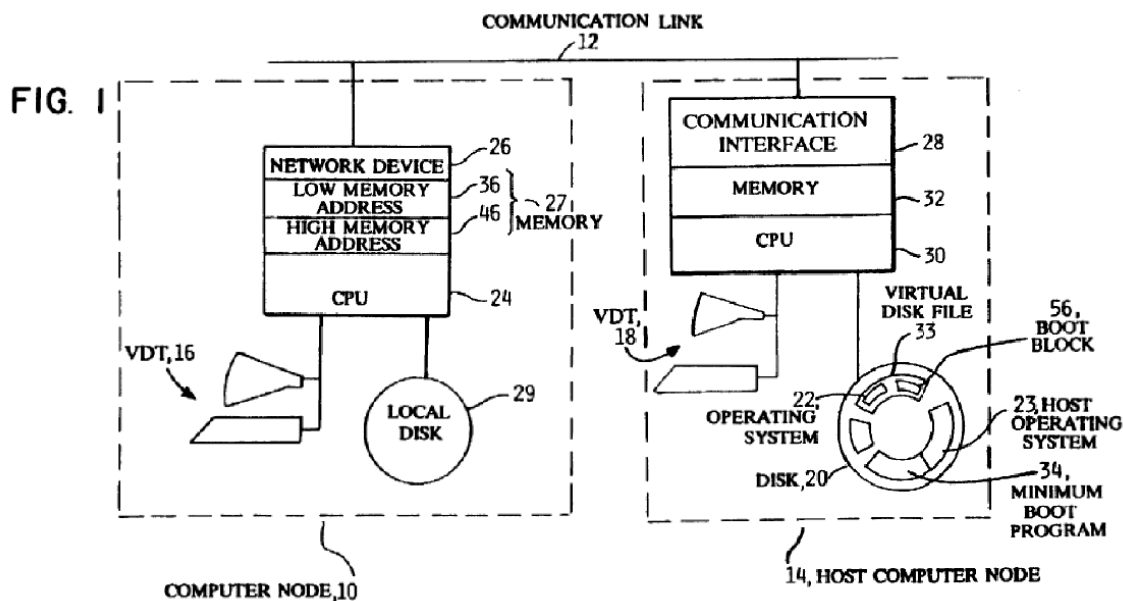


Figure 1 depicts a network having a computer node 10 that makes use of the host OS 23 and a minimum-boot program 34 of computer host node 14 via communications interface 28.

Flaherty describes Figure 1 at column 5, lines 1–21. It depicts a network having a computer node 10 that makes use of the host OS 23 and a minimum-boot program 34 of computer host node 14 via communications interface 28. At power up, processor 24 executes a power on self test (“POST”) sequence during which it attempts to find a boot device which contains a file with its operating system in it. If it does not find a boot device, the network device 26 broadcasts a request for an operating system over the network communications line 12. The network device 28 on the host 14 determines that the message is in broadcast mode and notifies CPU 30. Ex. 1006, 4:40–68. The CPU 30 determines a message to be a boot request and identifies an OS for the node 10. If it finds such an identification, it constructs a message by attaching the hardware address of the network device 26 of node computer 10 onto a minimum-boot program file 34 and places the message on the communications line 12. The communications device 26 of the node 10 receives the message and copies the minimum boot program into the node’s low memory addresses 36 of its memory 27. The CPU 24 then begins to execute the minimum-boot program. The minimum-boot program is then copied into the high addresses 46 of memory 27 and the POST sequence completes. Ex. 1006, 5:1–21.

C. Challenges Relying on Chang and Dent

Petitioner contends that claims 1–18, 24–30, 32, 35–53, 55, 56, 59 and 60 are obvious under 35 U.S.C. § 103 based on Chang and Dent, relying on the supporting testimony in the form of a Declaration by Mr. Stefano Righi (Ex. 1019). Pet. 9. Petitioner provides a detailed reading of these challenged claims on Chang and Dent, including references to the Righi Declaration. *Id.* at 11–47.

Patent Owner argues that this challenge should be denied because portions of Mr. Righi’s testimony are not discussed in the Petition itself. PO Resp. 6–29. Patent Owner correctly notes that a Petitioner must make its arguments in the Petition. We have only considered portions of the Righi testimony that relate to arguments presented in the Petition.

Except for claims 15, 16 and 50, as noted below, Petitioner’s detailed reading of the challenged claims on the references has established by a preponderance of the evidence that these claims are unpatentable based on Chang and Dent. The discussion below focuses on claims for which Patent Owner provides argument, albeit not supported by any declaration testimony. These arguments essentially contend that Petitioner did not sustain its burden to establish unpatentability.

1. Combining Chang and Dent

Petitioner argues that one of ordinary skill would have had a reason to combine Chang with Dent because Chang’s system can pre-boot remote diagnostics. Pet. 13 (citing Ex. 1003, 3:32–33). Dent provides specific details on how to implement diagnostics in a similar computing environment. *Id.* (citing Ex. 1004, Figure 3–2). Both references provide

similar solutions to the problem of providing communication capability before the OS is loaded. Both references disclose using a BIOS to establish a network connection before an operating system is active. *Id.* (citing Ex. 1019 ¶ 56).

According to Petitioner, Chang implements firmware that interfaces with a server management application (SMA) of a remote computer (e.g., workstation server) before a full operating system is active. *Id.* (citing Ex. 1003, 4:46–50; Ex. 1019 ¶ 56). Chang details how a remote server accesses, updates, manages, controls, and repairs a local computer. *Id.* (citing Ex. 1019 ¶ 42; Ex. 1003, 3:12–50). Dent describes a similar concept, i.e., after computer initialization upon power-on, establishing remote connection by initializing a network interface controller (NIC) in response to a boot error. *Id.* (citing Ex. 1004, Figure 5, item 560).

Patent Owner argues that there is no motivation to combine Chang and Dent. PO Resp. 6–7. According to Patent Owner, Petitioner’s analysis ignores fundamental differences in operation between the systems taught in the two references. PO Resp. 6. Chang establishes a connection with a server upon detection of a program in the PROM. Pet. 6–7 (citing Ex. 1003 at 6:55–57 (“The system BIOS detects the existence of a program in a ROM socket on a network interface card (step 35) and passes control to this program (step 37).”)). Dent establishes communication only when there is a boot error. Ex. 1004 at 3:4–8 (“to enable remote system level diagnostics through a privately or publicly accessible network in the event of a boot error”). Patent Owner argues that given these differences, there would have been no motivation to combine these references. PO Resp. 6–7.

We determine that the differences noted by Patent Owner would not dissuade one of ordinary skill from combining the teachings Chang and Dent. As noted above, Chang, during a boot sequence and prior to loading its operating system (OS), uses its BIOS to enable network communication. Dent describes a boot sequence that establishes an interconnection over a network for remote servicing of an electronic system upon experiencing a boot error. Thus, they address the same general problem and are similar in overall approach to solving that problem.

2. Independent Claims 1, 38, and 52

Independent claim 1 requires “loading and executing a network-enhanced software on said CPU on completion of at least a first portion of said start-up software.” The specification describes “early POST” 510 and the rest of POST 520. Ex. 1001, 6:38–58.

Petitioner reads claim 1 on Chang and Dent, noting that both of these references describe the limitations of claim 1. Petitioner argues that Chang’s BIOS detects the existence of a program in the ROM of a network interface card and then passes control to this program. Petitioner also argues that Dent executes BIOS software upon completion of at least a first portion of initial start-up software (ISUS). Pet. 14–17.

We agree with Petitioner’s understanding of Chang and Dent as they apply to the limitations of claim 1. Claim 1 merely recites that a “first portion” of software is executed. Petitioner demonstrates that at least some portion of software has been executed in both Chang and Dent before “network-enhanced software” is loaded and executed.

As to independent claims 1, 38, and 52, Patent Owner argues that Chang and Dent do not meet the “network-enhanced software” limitation because Chang’s operation always requires an operating system to be present before connecting to the network server. PO Resp. 7.

According to Patent Owner, Chang uses its mini-operating system to download the network communications software from the workstation storage medium (Paper No. 16 at 14), and the client and the server communicate in a sterile operating environment (*Id.* at 15). However, according to Patent Owner, “Chang requires an operating system to be present before connecting to the network server.” *Id.* at 29. Chang teaches “[a] suitable operating system kernel 25 is available from Hitech Business Solutions, Inc. of Wilmington, Del.” Ex. 1003 at 6:43–45. This “kernel” is located in Hitech’s X-DOS disk operating system for personal computers. HBS X-DOS 5 PC Disk Operating System User Manual, 1–1 & 5–10 (Aug. 8, 1995) (Ex. 2003).

Patent Owner further argues that under a proper construction of “network-enhanced software,” Chang’s operating system kernel does not satisfy the “network-enhanced software” required by claims 1, 38 and 52 of the ‘912 patent. PO Resp. 8.

As stated in our claim construction, we do not limit the term “network-enhanced software” as suggested by Patent Owner. Rather, we construe “network-enhanced software,” as used in claim 1, to encompass the kernel of an operating system.

Thus, we are not persuaded by Patent Owner’s arguments that Chang’s “kernel” does not meet the requirements of the “network enhanced

software” described by independent claims 1, 38 and 52. We conclude, based on Petitioner’s detailed read of these independent claims on Chang and Dent that Petitioner has established by a preponderance of the evidence that claims 1, 38, and 52 are unpatentable based on Chang and Dent.

3. Independent Claims 37 and 51

Claims 37 and 51 do not refer to “networked enhanced software,” as in claim 1. Rather, they recite a “second BIOS” supporting communication. Thus, at least as to this limitation, claims 37 and 51 are not as broad in scope as claim 1. Independent claim 37 is reproduced below.

37. A method of operating a computer, said computer including a central processing unit (CPU) and a network interface coupling said computer to a network, said CPU operating in at least two modes including a real mode and a protected mode, said method comprising:

executing a first basic input/output system (BIOS) on said CPU in real mode, said first BIOS performing a power on system test of said computer;

executing a second BIOS on said CPU in protected mode after execution of a portion of said first BIOS, said second BIOS supporting a network protocol stack; and

executing an operating system on said CPU in protected mode after execution of a portion of said second BIOS, said operating system software supporting user-level application software programs executing on said CPU.

Petitioner’s claim chart for claim 37 appears at pages 37–39 of the Petition. The claim chart points to a statement in the ’912 patent that use of real and protected modes was commonly known (Ex. 1003, 7:20–24). We credit this statement and find that those of ordinary skill would, at the time

of the invention, understand how to use real and protected modes. The claim chart also points to the subject matter of Melo (Ex. 1010, 4:10–14) as being known to one of ordinary skill in the art. Melo describes in detail transitioning between real and protected modes of operation of a CPU. Petitioner points to Bansen (Exhibit 1011, 1:43–50), which describes how UNIX operates in protected mode and directly controls its applications, as being known to one of ordinary skill. Petitioner also points to IBM (Ex. 1017), which describes switching from real to protected mode, as being known to one of ordinary skill. Petitioner also points to Bartek (Ex. 1012, 1:45–47), which explains operating a CPU running OS/2 in protected mode, as being known to one of ordinary skill. Petitioner relies upon Chang and Dent in much the same manner as with respect to claim 1. *Id.*

Patent Owner argues that the plain meaning of “BIOS” is a “basic input/output system.” PO Resp. 9 (citing Ex. 1003, 5:8). According to Patent Owner, there is nothing in the Chang reference that describes its operating system kernel as being a portion of a BIOS. Patent Owner notes that kernel is depicted and described as part of the PROM or ROM of the invention. *Id.* (citing Ex. 1003, 6:8–11, 6:30–38, Figure 2). Chang’s BIOS detects the presence of a program in a ROM socket and passes control to the program in the PROM. *Id.* at 6:55–59. Thus, according to Patent Owner, Chang’s operating system kernel is not part of BIOS, and cannot satisfy the “second BIOS” limitation of claims 37 and 51. Patent Owner argues that because the Board did not rely upon the Dent reference as separately teaching this limitation, the subject prior art combination lacks this limitation.

Petitioner argues, and we agree, that one of ordinary skill would consider Chang's firmware to be a "BIOS." Pet. 30 (citing Ex. 1019 ¶ 113). Petitioner also provides a detailed argument in its claim chart that Chang's operating system kernel is, in effect, a "mini operating system." Pet. 38 (citing Ex. 1003, 6:30–38). Patent Owner has not cited any testimony or other evidence to the contrary.

Patent Owner further argues that claims 37 and 51 require that the "first BIOS" execute in "real mode" and that Chang and Dent say nothing about a) operating BIOS in real mode; and b) operating a second, discrete BIOS in protected mode; and c) operating two BIOS programs sequentially in real and protected modes. PO Resp. 9–10.

Patent Owner correctly notes that Petitioner relies upon the knowledge of one of ordinary skill when reading Chang and Dent. This knowledge includes the subject matter described by Melo, Banset, IBM and Bartek. The ordinarily skilled person reading these references would appreciate that when a computer using an Intel 80386 or 80486 processor is first turned on, it begins operation in "real mode." Pet. 37; Ex. 1019 ¶ 53. Petitioner establishes that Chang meets the "second BIOS" limitation by offering the un rebutted Righi testimony which explains the knowledge of one of ordinary skill of real and protected modes, and that it would have been obvious to execute the second BIOS in protected mode, which was well-known at the time of the invention. Ex. 1019 ¶¶ 48–57. Patent Owner offers no testimony to counter the Righi testimony.

We find that, at the time of the invention, real and protected modes were commonly used modes of operation in Intel processors and that these

modes would have been known to those of ordinary skill. One of ordinary skill would have applied this knowledge along with the specific teachings of Chang and Dent to make the claimed invention. We therefore conclude that Petitioner has established by a preponderance of the evidence that claims 37 and 51 are unpatentable based on Chang and Dent.

4. Claims 12, 18, and 47

Claims 12 and 18 depend from claim 1. Claim 47 depends from independent claim 38 describing a “network-enhanced computer.” These claims describe plural software threads executed within a multi-tasking kernel. Claim 12 requires “said network-enhanced software multi-tasks a plurality of software treads.” The ’912 patent specification describes: “[T]he network-enhanced BIOS kernel 620 is preferably multithreaded, that is, it supports the apparently simultaneous execution of multiple tasks by switching the CPU control from one task to another according to which task has the highest priority.” Ex. 1001, 7:30–34.

Petitioner argues that one of ordinary skill would have found it obvious to execute a plurality of software threads on a multitasking kernel because Chang states that its kernel can be implemented using OS/2 or UNIX, which are multitasking kernels. Pet. 23–24.

Patent Owner argues that when we evaluated Petitioner’s challenge of claim 16, we agreed with Patent Owner that an ordinarily skilled person would not be motivated to design a multitasking kernel absent a teaching in Chang or Dent as to how such kernel would operate in conjunction with the other teachings of those references. Dec. to Inst., 23; PO Resp. 11. Patent Owner argues the same reasoning applies to these claims, which require a

multitasking kernel. PO Resp. 11. Patent Owner's argument does not take into account our Decision on Rehearing which added claim 16 to the review.

Petitioner has established that Chang explains that its kernel (claimed "network-enhanced software") is a mini OS. Pet. 23–24 (citing Ex. 1003, 6:55–64; Figure 2). Chang explains that this kernel 25 may be implemented using O/S 2 or UNIX. Pet. 23–24 (citing Ex. 1019 ¶¶ 86 and 89; Ex. 1003, 6:30–38).

We are persuaded that it would have been obvious to one of ordinary skill to execute a plurality of software threads on a multitasking kernel. Such an ordinarily skilled person would have been aware that multitasking threads significantly improve the efficiency of network-based operations such as remote diagnostics and maintenance (*id.*). Because UNIX and OS/2 were multitasking, the UNIX or OS/2 kernel in Chang would have suggested to one of ordinary skill to employ multitasking to improve efficiency.

Patent Owner argues Chang does not teach that its operating system kernel or its network communications software is implemented using OS/2 or UNIX, which support multitasking. PO Resp. 11. According to Patent Owner, in the absence of specific teachings, it would be wrong to assume that either Chang or Dent is a multitasking program that multitasks a plurality of software threads. In fact, according to Patent Owner, Chang teaches installing DOS on a workstation. Ex. 1003, 6:35–38. DOS is not a multitasking operating system. PO Resp. 11 (citing Ex. 2011, 3).

We disagree. We read references through the lens of one of ordinary skill in the art. Such a person, at the time of the invention, would have found it obvious to execute a plurality of software threads on a multitasking

kernel. *See* Pet. 24 (citing Ex. 1019 ¶ 89). Petitioner argues, and we agree, that one of ordinary skill would have been aware that multitasking threads significantly improve the efficiency of network-based operations such as remote diagnostics and maintenance. *Id.* The UNIX or OS/2 kernel in Chang suggests to one of ordinary skill to employ multitasking to improve efficiency. *Id.*

Patent Owner argues that even if Chang's operating system supports multitasking, this does not automatically equate to the programs running on top of that system being a multitasking program. According to Patent Owner, whether or not a program is able to multitask threads depends entirely on how the program is coded. PO Resp. 12 (citing Ex. 2009, 4). Patent Owner argues it is irrelevant to the platform where the program is running. *Id.* (citing Ex. 2009, 10). In short, Patent Owner argues nothing in the cited references teaches the multitasking required by these claims. PO Resp. 12.

We are not persuaded. Petitioner, with the supportive testimony of Mr. Righi, argues that Chang teaches multitasking. We find that one of ordinary skill would have understood and applied this very basic and well-known operation. Chang explains that its NES may include a kernel, which Chang refers to as a mini OS. Ex. 1003, 6:55–64; *see also id.*, Figure 2. Chang explains that this kernel 25 may be implemented using O/S 2 or UNIX. Ex. 1019 ¶ 86; Ex. 1003, 6:30–38.

We agree with Petitioner that one of ordinary skill would have understood that O/S 2 or UNIX based kernels are multitasking kernels. Ex. 1019 ¶ 86. It would have been obvious to execute a plurality of software

threads on a multitasking kernel. Ex. 1019 ¶ 89. One of ordinary skill would have been aware that multitasking threads significantly improve the efficiency of network-based operations such as remote diagnostics and maintenance. *Id.* The UNIX or OS/2 kernel in Chang would have suggested to one of ordinary skill to employ multitasking to improve efficiency in other contexts such as that disclosed in Dent. *Id.*

We conclude that Petitioner has demonstrated by its detailed read of these claims on Chang and Dent, supported by the declaration testimony that claims 12, 18, and 47 are unpatentable based on Chang and Dent.

5. Claims 13 and 48

Claims 13 and 48 recite that the network-enhanced software executes in protected mode. Petitioner argues “Chang teaches using a UNIX or OS/2 kernel,” which are “multitasking kernels that are necessarily executed in protected mode.” Pet. 24.

Patent Owner argues that whether a program is executed in protected mode is not dispositive as to whether a multitasking kernel is installed in a workstation. PO Resp. 12 (citing Ex 2009, 4).

Petitioner argues that the concept of real mode and protected mode fall within the general knowledge of one of ordinary skill because it implicates virtually every commercial PC at the time of the invention. Pet. 22. We agree, as noted above.

Patent Owner further argues that Chang teaches using DOS, which does not support multitasking. PO Resp. 12. While it is true that Chang teaches using DOS, it also teaches using OS/2, which uses multi-tasking. We therefore conclude that Petitioner has demonstrated by a preponderance

of the evidence that claims 13 and 48 are unpatentable based on Chang and Dent.

6. Claim 35

Claim 35 requires that the CPU operates in at least two modes, including a protected mode, as well as the steps of executing the NES and OS in protected mode, and terminating execution of the NES upon loading the OS.

Patent Owner argues that Petitioners reliance on Bartek as evidence of the general knowledge of an ordinarily-skilled person is misplaced. PO Resp. 13 (citing Pet. 36–37). Bartek teaches “[a]s part of the OS/2 2.0 Operating System 100, there is a MVDM manager 102 that manages the multiple virtual DOS machines that run under OS/2 2.0 in the V86 mode.” Ex. 1012, 6:48–51 (emphasis added). According to Patent Owner, a program preferably operating in protected mode, does not automatically ensure that the program will be executed in protected mode. PO Resp. 13 (citing Ex. 1011, 11: 8–20; Ex. 1012, 11:18–27, Fig. 7B).

We are not persuaded by Patent Owner’s argument. Petitioner’s reliance on Bartek is an example of what an ordinarily skilled person would know about using protected mode operation. We conclude that operating in protected mode was well within the knowledge of one of ordinary skill at the time of this invention, based at least on Bartek. One of ordinary skill would understand “preferably” to be a suggestion to do so.

In view of Petitioner’s reading of claim 35 on Chang and Dent we are persuaded by a preponderance of the evidence that claim 35 is unpatentable based on Chang and Dent.

7. Claims 14 and 49

Claim 14 requires the execution of the start-up software to be in real mode. Petitioner asserts that “every Intel-based PC which supports at least real and protected mode begins in real mode upon power on.” Pet. 24. According to Petitioner, as such, Chang and Dent necessarily begin POST in real mode. *Id.*

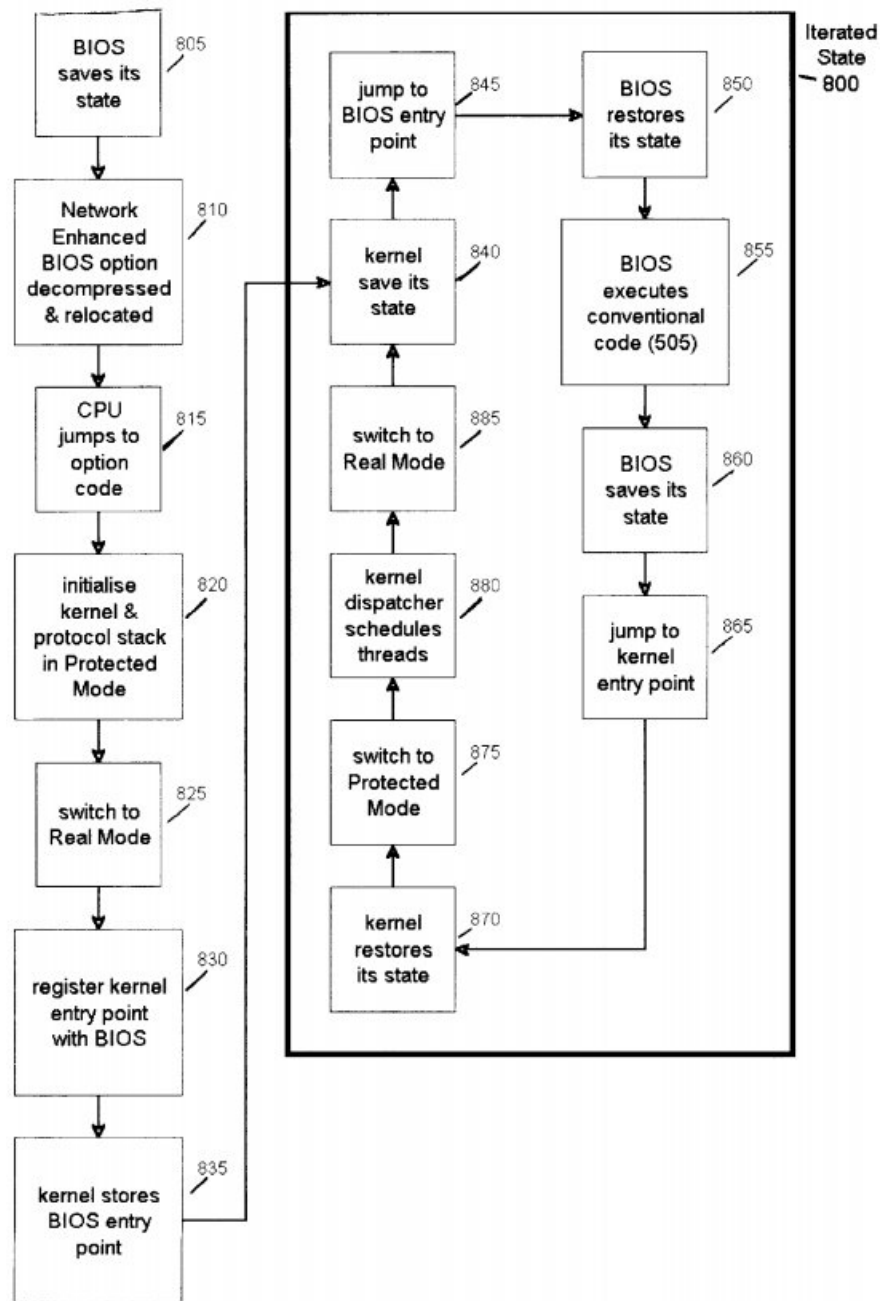
Patent Owner argues that there is no link between the processor mode upon power-on and the processor mode while executing a particular BIOS program, e.g., start-up software. PO Resp. 13 (citing Ex. 1019, 53).

We are not persuaded by Patent Owner’s argument. Claims 14 and 49 simply require the additional limitation of supporting at least real and protected mode and starting POST operation in real mode. Given that all Intel-based PC processors support both modes, we conclude that the real mode requirement of claims 14 and 49 would have been obvious to one of ordinary skill. We conclude that Petitioner has established based on a preponderance of the evidence that claims 14 and 49 are unpatentable based on Chang and Dent.

8. Claims 15, 16, and 50

Claims 15 and 50 recite “said start-up software and said network-enhanced software sharing control of said CPU after said CPU begins executing said network-enhanced software.” The ‘912 patent depicts and explains how this “sharing control” works in Figure 8 reproduced below:

Fig 8.



Ex. 1001, Figure 8.

Patent Owner points to iterated state 800 and argues that the '912 patent processor alternately executes the start-up program and the network-

enhanced software. Ex. 1001, 8:4–26. The processor may execute certain routines of the start-up program during which the control of the processor is with the start-up program. Some of the routines may require more time to finish, so the processor starts to execute threads of the network-enhanced software while waiting for the completion of the routines, during which the control of the processor passes to the network-enhanced software. When there is no ready thread to be executed, the processor goes back to execute the routines of the start-up program, during which the control of the processor passes to the start-up program again. *Id.* at 8:4–26, Figure 8 (items 865, 870, 875, and 880). The process repeats until all routines of the start-up program and all threads of the network-enhanced software are finished. *Id.* at 8:4–26, Figure 8 (Item 800). Such control sharing (iterated state 800) occurs after the processor begins executing the network-enhanced software. *See id.*, Figure 8 (Items 810, 815 and 820).

However, argues Patent Owner, there is no iterated state in either Chang or Dent. According to Patent Owner, Petitioner cannot equate Chang’s steps 33, 37 and 53, with this claim limitation. Thus, Patent Owner argues Chang’s BIOS and network communications software do not “share” the processor after the CPU begins executing the network communications software. Ex. 1003, Figure 3a (item 37). Instead, Figure 3b of Chang shows the processing steps after the execution of the network communications software. According to Patent Owner, as illustrated, control of the processor remains on the network communications software all the way through its completion. *Id.* at 6:58–7:22 and Figure 3b (items 39–51). Patent Owner argues that only after the completion does the control return to BIOS. *Id.* at

7:22–24, Figure 3b (Item 53). Patent Owner argues that Chang does not describe that control passes back to the network communications software afterward. Thus, Patent Owner argues the BIOS and the network communications software cannot share control of the processor, as recited in the challenged claim. Even if a UNIX kernel may allow operations in protected mode while other software is operated in real mode (*see* Pet. 25), Patent Owner argues there is no suggestion in either Chang and Dent that this is truly the case. Patent Owner argues Chang teaches the use of an operating system where no protected mode and no multi-tasking kernel are supported, i.e., DOS. Ex. 1003 at 6:35–38. Patent Owner argues it would be purely conclusion to argue that this prior art combination teaches control sharing after the execution of the network-enhanced software. PO Resp. 13–17.

Although Chang’s teachings are not limited to DOS, as suggested at the end of Patent Owner’s argument (PO Resp. 17), we are persuaded by this argument. The Petition does not demonstrate Chang or Dent teaches shared control after the CPU begins executing the network-enhanced software. Nor does it present any reasoning as to why it would have been obvious to do so. We therefore conclude that Petitioner has not established by a preponderance of the evidence that claims 15 and 50 are unpatentable based on Chang and Dent.

Claim 16 depends from claim 15 and states “wherein said network-enhanced software controls switching of the real mode and protected mode of said CPU.” . Petitioner’s presentation as to claim 16 does not cure the deficiency of claim 15, discussed above. We therefore conclude that

Petitioner has not established by a preponderance of the evidence that claim 16 is unpatentable based on Chang and Dent.

9. Claims 2, 3, 4, 39, 40, and 41

Claims 2, 3, and 4 require “a network protocol stack.” Claims 3 and 4 further require that stack to comply with industry standards. Figure 6 of the ’912 patent specification illustrates an embodiment of the network protocol stack. Petitioner points to Chang’s network interface connector (NIC) as satisfying this claim limitation. Pet. 17 (citing Ex. 1003, 4:10–17).

Patent Owner argues that unlike the term “protocol,” which refers to a single set of rules regulating how data is transmitted between two endpoints, the term “protocol stack” is an implementation of a computer networking protocol suite. PO Resp. 17 (citing Ex. 2001, 2). According to Patent Owner, Petitioner ignores the distinction. PO Resp. 17.

Patent Owner points to the testimony of Dr. Hugh Smith, in IPR2015-01133, who opined that the NetWare kernel protocol in Chang is operating system software. PO Resp. 17 (citing Ex. 2002 ¶ 207). Mr. Righi, in contrast, equates the NetWare kernel with the claimed protocol stack of the ’912 patent. Ex. 1019 ¶ 64.

We find that one of ordinary skill would have been capable of implementing the required protocol stack based on Chang. We are not persuaded by Patent Owner’s reference to the Smith testimony (PO Resp. 17–18) that there is an inconsistency sufficient to support Patent Owner’s argument that Petitioner has not carried its burden. It is not clear from Patent Owner’s argument why it would be not be possible for Chang’s kernel to carry out the function of a protocol stack as Mr. Righi’s testimony

suggests. The “inconsistent” testimony that Patent Owner points to does not suggest to us that one of ordinary skill would not have been able to implement the claimed protocol stack.

Patent Owner argues that Claim 2 also requires the network protocol stack to support “a driver for said NIC.” Petitioner suggests that, because Chang discloses an NIC and a network driver program, it also shows that “the protocol in use includes a network protocol stack supporting a driver for a NIC.” PO Resp. 18. Mr. Righi opines that “the fact that the NES facilitates the workstation-server communication demonstrates a network protocol stack.” Ex. 1019 ¶ 71. Patent Owner argues that the conclusion relies upon the assumption that Chang operates as a functioning DOS operating system. However, according to Patent Owner, most NIC vendors at the time supported drivers that were user-level programs written for standard DOS environments, like the functioning X-DOS operating system provided by Chang’s pristine operating system environment. *See* Ex. 2005, L2.

Mr. Righi states that Dent discloses a protocol that comprises a network protocol stack supporting a driver for a NIC. Ex. 1019 ¶ 71. He then notes that Dent’s software initializes a NIC (Ex. 1004, Fig. 5, item 560). We find that these facts support Mr. Righi’s opinion that “the fact that the NES facilitates the workstation-server communication demonstrates a network protocol stack” is sufficiently supported and we conclude that Petitioner has met its burden with respect to claim 2.

Claims 3 and 4 require that the network protocol stack supports a network driver layer, which complies with at least a subset of the ODI

specification, or with a subset of the NDIS specification. Petitioner relies on Mr. Righi's testimony that Netware makes use of drivers according to the ODI specification and LAN Manager makes use of drivers according to the NDIS specification; and therefore, Chang discloses a network protocol stack that supports a network driver layer, which complies with either the ODI or NDIS specifications. Pet. 18.

Patent Owner argues that both NetWare and LAN Manager for DOS provide user-level standard support for DOS environments, like the functioning X-DOS operating system provided by Chang's pristine OS environment. PO Resp. 18–19 (citing Ex. 2002, 207; Ex. 2006, 2). We are not persuaded by this argument and find Mr. Righi's testimony is sufficient basis for us to conclude that the limitations of claims 3 and 4 would have been obvious to one of ordinary skill. Ex. 1019 ¶ 73.

Patent Owner argues that as to claims 39, 40 and 41, Petitioner relies on arguments made for claims 2, 3 and 4 – arguments that fail for the same reasons as discussed above. PO Resp. 17–19 (citing Pet. 41). For reasons noted above with respect to those claims we conclude that Petitioner has established by a preponderance of evidence that claims 39, 40 and 41 are also unpatentable.

10. Claims 5 and 42

Claims 5 and 42 require that the network interface comprises a modem coupled through the CPU, the modem operating according to a protocol for communicating with the network. Petitioner relies on Dent's teaching of "initializing the modem" to meet this limitation. Pet. 32.

Patent Owner provides no substantive argument regarding this limitation.

These claims do not require any particular type of modem. We conclude that one of ordinary skill at the time of the '912 patent invention would have understood known network modems and protocols. Dent, read through the lens of one ordinarily skilled, describes a modem and its initialization. We find this to be sufficient to meet the limitations added by claims 5 and 42 and therefore conclude that Petitioner has established by a preponderance of the evidence that claims 5 and 42 are unpatentable based on Chang and Dent.

11. Claim 7

Claim 7 depends from claim 6 and requires that the notification recited in claim 6 comprise an SNMP alert message. Petitioner relies on Chang's discussion of SNMP as an optional protocol for the communication with the server. Pet. 19 (citing Ex. 1003, 6:19–20).

Patent Owner argues that Chang uses a functioning operating system, and it is of no moment that such operating system interfaces with an SNMP. PO Resp. 20.

We disagree. . Patent Owner does not argue that SNMP was not an established protocol. SNMP was erefore mentioned in Chang without further explanation because it was well known. We therefore conclude that Petitioner establishes by a preponderance of evidence that it would have been obvious to one of ordinary skill to include an SNMP alert message as required by claim 7. Claim 44 has a similar requirement as claim 7. We

conclude that Petitioner has established by a preponderance of the evidence that claim 7 is unpatentable based on Chang and Dent.

12. Claims 43 and 44

Claim 43 requires that a notification be sent indicating CPU system start-up to a network workstation before the CPU executes its operating system. Claim 44 depends from claim 43.

Petitioner points to Chang as meeting this limitation based on its description that in normal operation, as shown in the flowcharts of Figures 3a and 3b, boot is interrupted after the BIOS (in an MS-DOS environment, for example) is executed but before DOS executes. Pet. 42 (citing Ex. 1003, 6:49–52). Petitioner also points to Dent as satisfying this limitation: “If it is necessary to service a boot error in the electronic system before proceeding with the standard boot sequence, remote communications are established over the Internet as discussed above (Steps 420—430).” *Id.* (citing Ex. 1004, 5:8–12; *see also id.*, Figure 1, step 150, which depicts “booting an operating system” as part of the standard boot sequence).

Patent Owner argues that neither Chang nor Dent discloses the recited notification and that it is irrelevant that Chang focuses upon pre-boot operations (*see* Pet. 19) because these operations disclose a pristine and functional DOS operating system environment executing BIOS under DOS user instructions. PO Resp. 20–21 (citing Ex 1003, 6:30–33, 6:67–7:2).

We are not persuaded by Patent Owner’s argument. The passage of Chang quoted by Petitioner clearly describes that the boot process is interrupted before DOS executes. Communication thus occurs before the operating system is running. We conclude that Chang meets this claim

limitation and that claims 43 and 44 have been shown, by a preponderance of the evidence, to be unpatentable.

13. Claims 8 and 9

Claims 8 and 9 limit the SNMP alert message to relay specific information such as the cause of the CPO system reset (claim 8) or indicia that an OS has been loaded (claim 9). The Petition contends that Chang teaches adopting the SNMP standard to allow a local workstation and remote workstation to communicate with one another before an OS is active. Pet. 20 (citing Ex. 1019 ¶ 79; Ex. 1003, 6:1–8). According to Petitioner, SNMP traps are designed as an alert communication protocol. *Id.* (citing Ex. 1019 ¶ 79). Petitioner argues that one of ordinary skill, applying ordinary creativity, would find it obvious to leverage the SNMP protocol to alert a remote computer of particular errors and other events. *Id.*

Petitioner explains that with regard to claim 8, Chang contemplates providing “[r]emote software installation, distribution, metering and diagnostics.” Pet. 20 (citing Ex. 1003, 3:32–33). We agree with Petitioner that although Chang does not detail the contents of any messages, it would have been obvious to use SNMP to identify the cause of a CPU system reset. *Id.* (citing Ex. 1019 ¶ 80). Dent further demonstrates the need for allowing a remote technician to understand the cause for boot failures. *Id.* (citing Ex. 1004, 4:56–61).

With regard to claim 9, Chang sends acknowledgments when software (including an OS) is updated and loaded. *Id.* (citing Ex. 1003, 7:20–23; Ex. 1003, 6:12–20). In this respect, Chang teaches one of ordinary skill to send

an indication that an OS has been loaded. This shows that claims 8 and 9 would have been obvious. *Id.* (citing Ex. 1019 ¶ 81).

Patent Owner argues that neither Chang nor Dent teaches including “an identification of the cause of the CPU system reset” in the SNMP alert message, as required by claim 9. In fact, Patent Owner argues Petitioner admits “Chang does not detail the contents of any messages.” PO Resp. 21 (citing Pet. 20). Patent Owner contends Petitioner argues that “it would be obvious to use SNMP to identify the cause of a CPU system reset.” *Id.*

With regard to claim 8, Patent Owner argues that the SNMP protocol examples in Chang never define any identification altering the cause of the CPU system reset. According to Patent Owner, the two examples, a coldStart(0) trap and a warmStart(1) trap identified by Mr. Righi in his declaration, serve only to signify to a managing server that the agent (local workstation) is reinitializing. PO Resp. 21 (citing Ex. 1019 ¶ 80). They do not inform the managing server of “the cause of the CPU system reset.” *Id.* (citing Ex. 2013 at 5 and 6).

With regard to claim 9, Patent Owner argues that Chang does not teach the specific message required by the claim. Claim 9 recites “sending an SNMP alert message indicating said operating system has been loaded.” In our Decision to Institute, we stated that “Chang’s ‘ACK’ is provided to acknowledge ‘the update is successfully performed,’ but it does not indicate loading the operating system.” Dec. to Inst. 19. Patent Owner argues that, therefore, claim 9 is patentable over Chang and Dent. PO Resp. 22

Patent Owner is correct that Chang does not teach the specific messages required by the claim. However, the Petition persuasively

explains the knowledge of one of ordinary skill and such a person's familiarity with SNMP alert messages. Based on Petitioner's explanation regarding one of ordinary skill, we conclude that the particular recited messages would have been within the skill of such a person and therefore claim 9 would have been obvious in view of Chang and Dent.

We conclude that Petitioner has established by a preponderance of the evidence that claims 8 and 9 are unpatentable based on Chang and Dent.

14. Claims 10 and 45

Claims 10 and 45 require that the OS software is not executed without having established a network connection. According to Petitioner, Chang teaches one of ordinary skill to supplement its BIOS with firmware stored in ROM/PROM. Pet. 21 (citing Ex. 1003, Abstract). With the firmware in place, Chang's process follows the steps of Figures 3a and 3b. Step 41 ("initiation network server connection") occurs before step 55 ("complete personal computer boot"). Consequently, booting does not complete (e.g., loading OS) unless step 41 is completed. Pet. Reply 13 (citing Ex. 1019 ¶ 82).

Patent Owner argues that before Chang establishes a network connection it must detect a program in PROM. PO Resp. 22 (citing Ex. 1003, 6:55–64, Figure 3a (items 35 and 37), Figure 3b (items 39 and 41)). Assuming Chang's BIOS does not detect a PROM program, it would boot the computer and load an OS. *Id.* (citing Ex. 1003, Figure 3a (item 55)). That is, according to Patent Owner, Chang's operating system could be executed in the absence of the network connection.

Therefore, according to Patent Owner, Chang does not teach “said operating system will not be executed without the establishment by said network-enhanced software of a network connection with a remote workstation attached to said network,” as recited in claim 10. Claim 45 contains similar language (“preventing the execution of said operating system software prior to the establishment . . . of a network connection with a remote workstation attached to said network”) and Patent Owner argues that this claim is patentable over the prior art for the same reason. PO Resp. 22–23.

We are not persuaded by Patent Owner’s argument. Finding a scenario in which a particular operation does not occur does not negate the more general teaching suggesting the claimed limitation. The Petition establishes that Chang generally does not boot until step 41 is completed. Pet. 41 (citing Ex. 1019 ¶ 82). Thus, Chang teaches the limitations of claims 10 and 45. We conclude that Petitioner has established by a preponderance of the evidence that claims 10 and 34 are unpatentable based on Chang and Dent.

15. Claim 24

Claim 24 describes automatically entering into a diagnostic state when certain error conditions occur. According to Petitioner, Chang supports diagnostics. Pet. 27 (citing Ex. 1003, 3:32–33). Petitioner points to Dent’s Figure 5 as showing error conditions that lead to automatically executing remote access for diagnostic purposes. *Id.*

Petitioner also relies on Dent’s discussion of diagnostics described by Figure 6. Pet. 27–28. According to Patent Owner, the Figure 6 diagnostics

do not relate to “pre-determined error conditions” because diagnostics are carried out at power-on, regardless of error. PO Resp. 23–24. We are satisfied by Petitioner’s explanation (Pet. 27–28) that Dent automatically enters into a diagnostic state upon pre-determined error conditions.

Petitioner also relies on box 49 in Chang’s Figure 3b (Pet. 28), which “update[s] workstation files.” Ex. 1003, Figure 3b (Box 49). Further, Chang teaches “in step 49, the server sends any required workstation updates to the client and performs any assigned tasks under control from the SMA.” *Id.* at 7:10–12 (emphasis added). Updating files differs from the POST routines required by claim 24. Petitioner points to Dent’s Figure 5 as demonstrating predetermined error conditions (Ex. 1004, Figure 5, step 550, “boot error”) that lead to automatically executing remote access (*id.*, step 570 “boot error”) for diagnostic purposes (Ex. 1019 ¶ 100; Ex. 1004, 5:53–61). Dent explains that “[t]he types of boot errors may be inclusive of all possible types of boot errors, or restricted to a smaller range of boot errors.” Ex. 1004, 4:4–6. Petitioner also argues that Dent satisfies the need of repairing computers via “remote servicing in an automated fashion.” Pet. 25 (citing Ex. 1004, 2:29–30).

Patent Owner asserts that Dent does not teach “automatically entering into a diagnostic state.” PO Resp. 23–24. However, Petitioner notes that claim 24 does not preclude the intervening step of establishing communication between the time the error condition occurs and the time diagnostics begin. Pet. Reply 13 (citing Pet. 27, which in turn cites Ex. 1019 ¶ 100; Ex 1004, 5:53–61, Figure 5 (steps 550 and 570)).

We agree that claim 24 does not preclude additional steps. Further, in view of Chang, Dent’s teaching of automatically creating remote access for diagnostics when a boot error occurs renders claim 24 obvious. Thus, we conclude that Petitioner has established by a preponderance of the evidence that claim 24 is unpatentable based on Chang and Dent.

16. Claims 25 and 55

Claims 25 and 55 describe “additional software” to be downloaded. These claims do not limit the additional software to being a second portion of the start-up software, as in claim 11. Petitioner refers to its analysis of claim 11, where it argued that Chang combined with Dent discloses downloading addition software programs. Patent Owner does not respond to claims 25 and 55 being broader than claim 11.

Petitioner argues that Chang discloses claims 25 and 55 by describing updating and repairing critical software systems in order to transfer new files to the client prior to boot. Pet. Reply 14 (citing Ex. 1003, 4:65–68; 5:21–26; Ex. 1019 ¶ 83).

We agree with Petitioner’s analysis and conclude that Petitioner has established by a preponderance of the evidence that claims 25 and 55 are unpatentable based on Chang and Dent.

17. Claims 26 and 56

Claim 26 recites that “said remote workstation may further perform the steps of controlling power on system test (POST) routines.” Claim 56 has similar language.

Petitioner argues, with the support of Mr. Righi, that “Chang’s remote workstation updates workstation files, it eventually determines when to

resume the POST routine of Chang’s BIOS.” Pet. 28 (citing Ex. 1019 ¶ 102). Petitioner further argues that the “controlling” of claims 26 and 56 includes controlling the timing of a POST routine. Pet. Reply 14 (citing Ex. 1019 ¶ 102).

Patent Owner argues that “[o]nce control is returned, it is impossible for Chang’s remote workstation to perform any operations, including POST routines. PO Resp. 24–25 (citing Ex. 1003, 7:22–24 and Figure 3a, item 55).

We agree with Petitioner that Chang does not teach that POST cannot resume. We find no support for Patent Owner’s argument in this regard. According to Patent Owner, Chang states that “any preboot activity . . . take place . . . under control of the SMA.” PO Resp. 26 (citing Ex. 1003, 4:1–6, Figure 1 (item 11c)). Patent Owner argues that the SMA is not the network communications software, but a server management application that resides on the server. While we agree that the SMA is not network communication software, we are not persuaded by Patent Owner’s argument. Chang discloses “controlling power on system test (POST) routines.” We credit Mr. Righi’s declaration testimony that determining when to resume the POST routine is an example of controlling the POST routine. Ex. 1019 ¶ 102. We conclude that Petitioner has established by a preponderance of the evidence that claims 26 and 56 are unpatentable based on Chang and Dent.

18. Claim 27

Claim 27 recites “said remote workstation may force said computer into a diagnostic state.” Petitioner relies in part on arguments made with respect to claim 24.

Patent Owner takes issue with Mr. Righi's opinion that some errors trigger a diagnostic state (server errors) and others do not. PO. Resp. 25 (citing Ex. 1019 ¶ 100; Ex. 1004, Figure 5). Patent Owner notes that claim 27 further requires "said network-enhanced software is available for further communication with said remote workstation." But, according to Patent Owner, Chang states that "any preboot activity . . . take[s] place . . . under control of the SMA." PO Resp. 26 (citing Ex. 1003, 4:1–6, Figure 1 (item 11c)). Patent Owner argues that the SMA is not the network communications software, but a server management application that resides on the server. PO Resp. 26.

Petitioner notes that Patent Owner's argument contradicts its argument with respect to claim 24 that "Dent's diagnostic recovery function cannot be enabled without . . . establish the [remote] communications." See PO Resp. 23–24). Petitioner argues, and we agree, that regardless, Dent teaches establishing remote communication in response to a severe failure so the remote server can control the local computer. Pet. Reply 15 (citing Ex. 1004, Fig. 5 (elements 550-570); Ex. 1019 ¶ 100). We therefore conclude that Petitioner has shown by a preponderance of the evidence that claim 27 is unpatentable based on Chang and Dent.

19. Claim 28

Claim 28 recites "said data further being transmitted from said computer to said remote workstation." Petitioner cites Dent's and Chang's teaching of remote access as fulfilling this limitation. Pet. 34–35 (citing Ex. 1004 at 4:56–63 and Ex. 1003 at 3:21–22).

Patent Owner argues that both Dent and Chang are silent as to data being further transmitted. PO Resp. 26–27.

We find that both Chang and Dent each teach remote access to a workstation hard drive. *See, e.g.*, Ex. 1003, 3:21, 3:32; Ex. 1004, 4:56–63. We find that the teaching of remote access to data of a first computer by a second computer would have suggested to one of ordinary skill transmitting data between two computers. *See* Ex. 1019 ¶ 104. We therefore find that Petitioner has established by a preponderance of the evidence that claim 28 is unpatentable based on Chang and Dent.

20. Claim 29

Claim 29 requires “said remote workstation transmitting data to said computer, said data being written to a direct access storage device locally coupled to said CPU.” Petitioner relies on Dent’s teaching that the computer technician may be able to analyze and correct an error. Pet. 35 (citing Ex. 1004, 4:64–66).

Patent Owner argues that Dent is silent regarding data written to storage in the local computer. Patent Owner also notes that although Chang teaches that “workstation files are updated” (Ex. 1003, 7:6–9, Figure 1 (item 17a)), Chang does not make clear where those workstation files are stored. PO Resp. 27.

We agree with Petitioner that Chang explains that workstation files include boot files. Ex. 1003, 2:67–3:1 (describing “updates to critical boot files prior to workstation boot by transferring those files from the server”). Chang also teaches that the workstation’s own operating system may be rescued. *Id.* at 4:46–50.

Patent Owner provides no evidence to refute Mr. Righi's opinion that the reference "suggests the ability to write data to a local storage device." Ex. 1019 ¶ 105 (citing Ex. 1004, 4:64–66, 5:59–61). We find that testimony persuasive. We therefore find that Petitioner has established by a preponderance of the evidence that claim 29 is unpatentable based on Chang and Dent.

21. Claim 30

Claim 30 requires "downloading from a remote workstation to said computer portions of said start-up software or said network-enhanced software, whereby such start-up software and network-enhanced software will be executed upon the next occurrence of system start-up."

Petitioner relies on its argument made with respect to claim 11. Pet. 35. According to Petitioner, Chang combined with Dent discloses downloading additional software programs. Claim 11 is narrower in scope than claim 30, requiring "said operating software not executing until said second portion of said start-up software has been downloaded and executed" which is not a limitation of claim 30.

Patent Owner's argument that the Board has decided not to institute a review of claim 11 does not take into account our Rehearing Decision adding claim 11 to the trial. Dec. on Reh'g.

Patent Owner also argues that Chang does not teach the recited software taking effect "upon the next occurrence of system start-up." PO Resp. 28. However, after Mr. Righi analyzed Chang's disclosure of recovering from damage to the client workstation's boot or operating system, Mr. Righi concluded that, "As a default rule, once software is

updated, the updated version will [be] executed upon the next system startup.” Ex. 1019 ¶ 106 (citing Ex. 1003, 5:11–15). We credit this testimony.

Given Mr. Righi’s testimony, we conclude that Petitioner has established by a preponderance of the evidence that claim 30 is unpatentable based on Chang and Dent.

22. Claim 32

Claim 32 requires the start-up software include an “early POST portion” and a “remaining portion of POST.” Petitioner relies on Chang’s Figure 3a for these limitations. Petitioner equates box 33 to the “early POST portion,” and Box 55 to the “remaining portion of POST.” Pet. 29.

Patent Owner argues that Chang does not mention that its BIOS includes “an early POST routine ... and a remaining portion of a POST.” PO Resp. 29. Additionally, Patent Owner argues, Chang teaches that “the system BIOS is executed (step 33) in a manner well known in the art.” Ex. 1003, 6:53–55. Nowhere, according to Patent Owner, does Chang mention that its BIOS includes “an early POST routine . . . and a remaining portion of POST” as required by this claim. Further, the BIOS execution in the art does not include “executing said early POST prior to said CPU’s loading of said network-enhanced software,” and “executing said remaining portion of POST after loading of said network-enhanced software.” PO Resp. 29.

Petitioner argues that Chang discloses an “early POST portion” and a “remaining portion of POST” in both steps 33 and 35 of Figure 3a. Pet. Reply 18. This includes executing BIOS in a well-known manner, detecting the presence of a PROM program, and passing control to the PROM

program. *Id.* (citing Ex. 1003, 6:52–55). This “well-known manner” encompasses POST, according to Mr. Righi. (Ex. 1019, ¶37). Chang teaches a “remaining portion of POST” by performing a normal boot process after the PROM returns control back to the workstation. (Ex. 1003, 7:20–24). Mr. Righi opines that step 55 of Chang’s Figure 3a maps to the 912 Patent’s Figure 4, items 413 and 414, which is the remaining portion of POST needed to perform a bootstrap function to load the OS. Patent Owner offers no explanation why, in view of the foregoing explanation, these limitations are not met.

We are not persuaded by Patent Owner’s argument. Petitioner points out that Chang discloses an SMA (server application) that controls the mini OS of a local computer. Pet. Reply 15 (citing Ex. 1003, 7:30-36). Petitioner points out that Patent Owner’s argument (PO Resp. 26) misconstrues Petitioners’ point that the mini OS is the network enhanced software that is controlled by the remote SMA. Pet. Reply 15–16.

We find Petitioner’s discussion of the early POST portion of Chang persuasive that Chang describes this claim limitation. Pet. Reply 18. When discussing Box 33, Chang discloses that “the system BIOS is executed (step 33) in a manner well known in the art.” Ex. 1003, 6:53–55.

Therefore, we conclude that Petitioner has established by a preponderance of the evidence that claim 32 is unpatentable based on Chang and Dent.

23. Claims 59 and 60

Claim 59 requires that “said network protocol stack is implemented by software stored in nonvolatile memory.”

We note Petitioner’s claim chart entry for claim 59 at page 46 of the Petition. Patent Owner argues that Petitioner points to nothing in the art describing “said network protocol stack.” PO Resp. 29–30

We noted in our discussion of claims 2, 3, and 4, above, that Petitioner sufficiently established that Chang and Dent teach a protocol stack. For the same reasons, we are not persuaded by Patent Owner’s argument regarding claims 59 and 60.

We adopt Petitioner’s claim chart at Petition page 46 and conclude that Petitioner has established by a preponderance of the evidence that claims 59 and 60 are unpatentable based on Chang and Dent.

24. Claim 36

Claim 36 further requires “retaining a portion of said network-enhanced software in memory after loading of said operating system software.”

Petitioner notes that claim 36 requires retaining in memory, not executing from memory. Pet. 29. According to Petitioner, one of ordinary skill would have known that the BIOS (as well any loaded Option ROM) is retained in memory after loading of the OS. *Id.* (citing Ex. 1019 ¶ 111). Chang demonstrates the loading and executing the NES (e.g., firmware) from a ROM/PROM. *Id.* (citing Ex. 1003, Figure. 3a (items 35 and 37)). Further according to Petitioner, Chang does not mention removing loaded firmware from memory. *Id.* (citing Ex. 1019 ¶ 111). This suggests to Petitioner that the NES is retained in memory after loading the OS included in step 55 of Figure 3a. Thus, according to Petitioner, Chang demonstrates the loading and executing the NES (e.g., firmware) from a ROM/PROM. *Id.*

Patent Owner does not provide argument specific to this claim. We adopt Petitioner's explanation at Petition page 29 and conclude that Petitioner has established by a preponderance of the evidence that it is unpatentable based on Chang and Dent.

25. Claims 17 and 53

Claims 17 and 53 require that the NES allows a remote workstation coupled to said computer network to access said computer. Both Chang and Dent teach this limitation.

We adopt Petitioner's claim chart entry for claim 17 at Petition page 33, on which Petitioner relies for both of these claims. Patent Owner does not provide any substantive argument with respect to this limitation.

We conclude that Petitioner has established by a preponderance of the evidence that claims 17 and 53 are unpatentable based on Chang and Dent.

26. Claims 6, 11 and 46

Although Patent Owner did not specifically argue claims 6, 11, and 46 in its Patent Owner Response, we have reviewed Petitioner's reading of these claims on Chang and Dent. We adopt Petitioner's explanation at Petition page 19 with respect to claim 6. We adopt Petitioner's explanation at Petition page 21–22 and its claim chart entry at Petition pages 32–33 with respect to claim 11. We adopt Petitioner's claim chart entry at Petition page 43 (referring to the entry for claim 11) with respect to claim 46.

D. Challenges based on Chang, Dent, and Bizzarri

Petitioner provides a detailed read of claims 19, 21–23, 28, 29, and 54 on Chang, Dent and Bizzarri at Petition pages 47–50, which we adopt. With

respect to claims 19, 21–23, 28, 29, and 54, Petitioner cites Bizzari for its teaching of various details related to preboot diagnostic techniques, including emulating a keyboard and screen.

1. Motivation to Combine

Patent Owner argues that there is no motivation to combine the teachings of these references based on Petitioner ignoring “design differences” of the three references. PO Resp. 30. These design differences allegedly include Chang requiring a functioning operating system, while neither Dent nor Bizzarri requires an operating system. Patent Owner notes that each of the references connects to a network in a different manner. *Id.* Finally, Patent Owner notes that Chang communicates upon a detection of a program in a PROM. *Id.* (citing Ex. 1003, 6:55–63). That is not the case in either Dent or Bizzarri. *Id.* (citing Ex 1004, 4: 56–61 and Figure 3–2 (350), Ex. 1008, 6: 20–25 and Figure 1 (13)).

We do not regard these design differences to be sufficient to discourage one of ordinary skill from combining the teachings for which these references are cited. These references fundamentally relate to establishing communication before fully booting an operating system.

2. Claim 19

Claim 19 requires that the network-enhanced software forward output normally directed to a video controller on the computer to the remote workstation. The claimed network-enhanced software is the program residing in the local computer and establishing communications with the remote workstation. *See* Ex. 1001, claims 1 and 17.

Petitioner relies on Bizzarri's teaching that "after slave kernel 22 is downloaded and tested, an interface at the diagnostic PC is provided emulating the keyboard, screen, and floppy drive of the failed computer." Pet. 48 (citing Ex. 1008, 7:25–35).

Patent Owner argues that the slave kernel 22 in Bizzarri does not establish the connection with a network server, per se. Rather, it is a program downloaded from the network server after the connection has been made. PO Resp. 31 (citing Ex. 1008, 7:8–10). According to Patent Owner, Petitioner failed to prove that the putative "enhanced network software" in Bizzarri outputs to a network server after the same software establishes communications with the network server. PO Resp. 31.

Petitioner argues in reply that Bizzarri teaches a pre-boot kernel (e.g., slave kernel) to perform the claimed "forwarding" functionality of the network-enhanced software. Pet. Reply 19 (citing Ex. 1008, 7:25–35; Ex. 1019 ¶ 132). Petitioner contends that Chang already suggests the functionality of claim 19 (Ex. 1019 ¶ 131 (citing Ex. 1003, 2:63–67)), and Bizzarri merely provides additional detail. Pet. Reply 19. Thus, according to Petitioner, the fact that Bizzarri's slave kernel may be downloaded is irrelevant when considering the combination of Bizzarri and Chang. Pet. Reply 19.

We agree with Petitioner that the core functionality is described by Chang (Pet. Reply 19) and therefore conclude that Petitioner has established by a preponderance of the evidence that claim 19 is unpatentable based on Chang, Dent, and Bizzarri.

3. Claims 21 and 54

Claim 21 depends from claim 17 and requires that the network-enhanced software permits the remote workstation to input messages simulating the effect of keystrokes. Claim 54 depends from claim 53 and has similar language.

Petitioner relies upon Bizzarri's emulation of a keyboard, screen and floppy drive of the failed computer. Pet. 48 (citing Ex. 1008, 7:25–33).

Patent Owner argues that the Bizzarri emulation is insufficient to meet the limitations of claims 21 and 54 because Petitioner does not prove that Bizzarri's network-enhanced software permits a network server to input messages after the same software establishes communications with the network server. PO Resp. 32.

We are not persuaded by Patent Owner's argument with respect to claims 21 and 54. Bizzarri provides the diagnostic PC with an emulated keyboard, screen, and floppy drive of the failed computer. We conclude that the Bizzarri emulation would sufficiently suggest to one of ordinary skill simulating the effect of keystrokes as required by these claims. Therefore Petitioner has established by a preponderance of the evidence that claims 21 and 54 are unpatentable based on Chang, Dent, and Bizzarri.

3. Claim 22

Claim 22 requires "said messages are received by a remote procedure call (RPC) server." Petitioner relies on its arguments made with respect to claims 20 and 21. Pet. 49. The Board decided not to institute a review on claim 20 because Petitioner failed to explain "how to apply the alleged widely used RPC to Chang or Dent." Dec. to Inst. 24. Thus, according to

Patent Owner, for at least for this reason, claim 22 reciting the same “RPC” requirement should be patentable.

Patent Owner is correct in that trial was not instituted for claim 20, on a challenge based on Chang and Dent. We now consider a challenge based on Chang, Dent, and Bizzarri to claim 22.

Petitioner argues that Mr. Righi explains that Bizzarri discloses emulating a keyboard without providing a specific implementation and RPC is an obvious mechanism by which to achieve keyboard emulation. Pet. Reply 19 (citing Ex. 1019 ¶ 134). Mr. Righi notes that RPC is designed in systems that have master and slave objects. Ex. 1019 ¶ 97 (citing Ex. 1016, 3:52–54 (“RPC Clients are also created in the server to allow the server to communicate with a master object’s slaves.”)). There is no counter testimony on this point, and we credit Mr. Righi’s testimony. We adopt Petitioner’s claim chart entries as to claim 22 (referring to claims 20 and 21) at Petition page 49. We therefore determine that Petitioner has established by a preponderance of the evidence that claim 22 is unpatentable based on Chang, Dent, and Bizzarri.

4. Claim 23

Claim 23 requires “said remote workstation displays messages and system errors reported by said start-up software executing on said CPU.” Petitioner relies on Bizzarri’s teachings of “on-screen message or icon to alert the technician to boot the host software and initiate diagnosis and repair for the offending PC.” Pet. 49 (citing Ex. 1008, 6:42–58).

Patent Owner argues that on-screen messages are insufficient. According to Patent Owner, Petitioner must also prove that Bizzarri’s start-

up software reports system errors to a network server for displaying the messages. PO Resp. 33.

Petitioner points out that Bizzarri explains how an “E-BIOS at the failed PC signals for remote service.” Pet. Reply 20 (citing Ex. 1008, 6:51). This “signaling” is the reporting of a system error. This signaling results in an “on-screen message.” *Id.* (citing Ex. 1008, 6:51–53).

Patent Owner offers no countervailing evidence. We adopt Petitioner’s claim chart entry for claim 23 at Petition page 49 and conclude that Petitioner has established by a preponderance of the evidence that claim 23 is unpatentable based on Chang, Dent, and Bizzarri.

5. Claim 28

Claim 28 depends from claim 17 and requires causing data to be read from a direct access storage device locally coupled to said CPU, said data further being transmitted from said computer to said remote workstation.

Patent Owner argues that Petitioner does not point to any prior art teaching that the read data is further transmitted to a network server. Petitioner points to the following portions of the references. Pet. 49.

Reference	Passage(s)
Bizzarri	“accessing and modifying code and data on read/write memory devices and mass storage devices in the first computer through activity at the second computer. (Ex. 1008, p.6, 4:41–43).”
Chang	Ex. 1003, 2:47–52; 3:21–22.
Dent	Ex. 1004, 4:56–63.

Petitioner did not address this issue in its Petitioner Reply. Petitioner has not explained how any of the cited passages describe what claim 28 requires. For example, Bizzarri teaches accessing and modifying code and data. It does not specifically describe transmitting data from direct access storage to the remote workstation. *See* Pet. 49. We therefore conclude that Petitioner has not established by a preponderance of the evidence that claim 28 is unpatentable based on Chang, Dent, and Bizzarri.

6. Claim 29

Claim 29 requires “said remote workstation transmitting data to said computer, said data being written to a direct access storage device locally coupled to said CPU.”

Patent Owner argues that Petitioner does not explain how the claim limitation is satisfied by the combination of Chang, Dent, and Bizzarri. PO Resp. 34.

The Petition points to two passages, one in Dent and One in Chang. Pet. 35. The Dent passage indicates that a computer technician may be able to analyze an error and correct the problem remotely. Pet. 35 (citing Ex. 1004, 4:64–66). The Chang passage states that an attempt is made to connect the workstation to the network and if successful, workstation files are updated. Pet. 35 (citing Ex. 1003, 7:6–9).

The Petition points to a passage of Bizzarri without explanation (Pet. 49–50 (citing Ex. 1008, 4:41–43)) along with a repeated reference to the previous citations to Chang and Dent. Petitioner does not address how these passages meet the limitations of claim 29. We therefore conclude that

Petitioner has not established by a preponderance of the evidence that claim 29 is unpatentable based on Chang, Dent, and Bizzarri.

E. Challenges based on Chang, Dent, and Aegis

Dependent claim 33 depends directly from claim 1, and dependent claim 34 depends from claim 33. Petitioner provides a detailed analysis of claims 33 and 34 with respect to Chang, Dent, and Aegis. Pet. 50–52. Petitioner cites Aegis for its teaching of carrying out authentication in a computer environment, arguing that Aegis modifies a boot process so that all executable code, except for a very small section of trusted code, is verified prior to execution by using a digital signature. Pet. 51 (citing Ex. 1019 ¶ 140). Petitioner argues that Chang supports authentication and thus Chang’s teaching of SNMP would have suggested to one of ordinary skill that the requirements of claims 33 and 34 would have been obvious. Pet. 50–52.

Patent Owner argues that because SNMP is “optional” in Chang, it would be “pure hindsight” to conclude that Chang’s teaching of SNMP renders claims 33 and 34 obvious. PO Resp. 34. We disagree.

We read Chang as one of ordinary skill would read it. SNMP is described, whether required or optional. We find that one of ordinary skill would not have disregarded a teaching because it is labeled “optional.”

Patent Owner also challenges Petitioner’s reliance on Aegis. According to Patent Owner, Petitioner’s Declarant, Mr. Righi, opines “Aegis shows that each message should be authenticated between a client and server in a preboot environment.” Pet. 51. However, according to Patent Owner, Aegis teaches integrity checks of executable codes between layers during its boot process, rather than messages between a client and a server via a

network. PO Resp. 35. We are not persuaded by this argument. AEGIS demonstrates authentication in a computer environment. We do not disregard this teaching because the authentication is not specific to a message received from a network. An obviousness inquiry is based on what a reference as a whole teaches one of ordinary skill. Aegis establishes that it was known to use authentication in a computer environment. We conclude that based on Aegis, it would have been obvious to an ordinarily skilled artisan to implement authentication as claimed. We adopt Petitioner's explanation at Petition pages 50–52 and conclude that Petitioner has established by a preponderance of the evidence of record that claims 33 and 34 are unpatentable based on Chang, Dent, and Aegis.

F. Challenges based on Flaherty and Chang (Claims 57 and 58)

Claims 57 and 58 differ from other independent claims by requiring enhanced software to establish a remote connection and then redirecting a BIOS disk services interrupt to access boot data from a remote workstation. Ex. 1019 ¶ 144. In addition, claims 57 and 58 limit the NES to be implemented as a kernel.

Petitioner provides a detailed analysis of claims 57 and 58 with respect to Flaherty and Chang. Pet. 52–57. Petitioner argues that Flaherty describes virtually the same system as that claimed using BIOS disk services interrupt (e.g., interrupt 13). Pet. 53 (citing Ex. 1006, 6:30–34). According to Petitioner, Flaherty discloses software that establishes a remote connection. Ex. 1006, 4:10–15. This software, which is a part of Flaherty's "minimum boot program" is chained to the POST program (e.g., ISUS). *Id.* at 4:15–16.

Petitioner acknowledges that Flaherty does not explicitly state that its software for establishing a remote connection is implemented with a kernel, but argues that the use of a kernel to connect remotely in a preboot environment is taught by Chang. Pet. 53 (citing Ex. 1003, Abstract). Chang states that using a kernel in this manner creates “a pristine operating environment.” Ex. 1003, 7:30–32. Given Chang’s encouragement, Petitioner argues that it would have been obvious to implement Flaherty using Chang’s kernel. Ex. 1019 ¶ 146. Chang’s kernel creates a “sterile” environment that allows a remote workstation to provide anti-virus protection. Pet. 53 (citing Ex. 1003, 3:6–11, 5:22–26).

Patent Owner argues that Flaherty does not run a “start-up program” because it does not comprise “an operating system bootstrap program.” PO Resp. 36–37. Patent Owner argues that to any skilled person, an operating system bootstrap program is a program that initializes an operating system by, e.g., using of BIOS services interrupt, such as INT 19. PO Resp. 37 (citing Ex. 2014, 7–1). Patent Owner cites three passages of Flaherty in support of its argument. *Id.*

Independent claim 57 requires “a start-up software, at least a portion of said start-up software being stored within said nonvolatile memory.” It does not require any particular type of start-up software. We construed “initial start-up software” to be “a set of instructions that initializes a computer system.” Dec. to Inst. 9–10 and Section II.A. of this Decision above. Patent Owner acknowledges that Petitioner relies on Flaherty as teaching a minimum-boot program. PO Resp. 40. As such, we are not persuaded by Patent Owner’s argument with regard to claim 57.

With regard to claim 58, both Petitioner and Patent Owner rely on their respective arguments made as to claim 57. As such, we are not persuaded by Patent Owner's argument with respect to claim 58.

We conclude that Petitioner's read of claims 57 and 58 on Flaherty and Chang to be sufficient to establish by a preponderance of evidence the unpatentability of these claims.

III. CONCLUSIONS

We conclude that Petitioner has established by a preponderance of the evidence of record that certain claims of the '912 patent are unpatentable as summarized in the table below:

Reference(s)	Basis	Claim(s) unpatentable
Chang and Dent	35 U.S.C. § 103	1–14, 17, 18, 24–27, 30, 32, 35–49, 51–53, 55, 56, 59 and 60
Chang, Dent and Bizzarri	35 U.S.C. § 103	19, 21–23, and 54
Chang, Dent, and Aegis	35 U.S.C. § 103	33 and 34
Flaherty and Chang	35 U.S.C. § 103	57 and 58

Petitioner has not established unpatentability of claims 15, 16, 28, 29 and 50 by a preponderance of the evidence

IV. ORDER

For the reasons given, it is:

ORDERED that claims 1–14, 17–19, 21–27, 30, 32–49, and 51–60 are unpatentable; and

FURTHER ORDERED that, because this is a final written decision, parties to the proceeding seeking judicial review of the decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.

PETITIONER:

Vivek Ganti
Gregory Ourada
HILL, KERTSCHER & WHARTON, LLP
vg@hkw-law.com
go@hkw-law.com

PATENT OWNER:

Christopher Frerking
chris@ntknet.com