

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SONY INTERACTIVE ENTERTAINMENT LLC,
Petitioner,

v.

BOT M8, LLC,
Patent Owner.

IPR2020-00726
Patent 8,112,670 B2

Before KALYAN K. DESHPANDE, LYNNE E. PETTIGREW, and
AMBER L. HAGY, *Administrative Patent Judges*.

DESHPANDE, *Administrative Patent Judge*.

JUDGMENT
Final Written Decision
Determining All Challenged Claims Unpatentable
35 U.S.C. § 318(a)
Denying Patent Owner's Motion to Exclude Evidence
37 C.F.R. § 42.64

I. INTRODUCTION

A. Background

Sony Interactive Entertainment LLC (“Petitioner”) filed a Petition requesting *inter partes* review of claims 1–5 of U.S. Patent No. 8,112,670 B2 (Ex. 1001, “the ’670 patent”). Paper 1 (“Pet.”). Bot M8, LLC (“Patent Owner”) filed a Preliminary Response. Paper 9 (“Prelim. Resp.”). With our authorization, Petitioner thereafter filed a Reply (Paper 11 (“Reply”)) and Patent Owner filed a Sur-reply (Paper 12 (“Sur-reply”)) to address issues involving 35 U.S.C. § 325(d).

On October 6, 2020, we instituted *inter partes* review of claims 1–5 of the ’670 patent. Paper 13 (“Dec.”). After institution, Patent Owner filed a Patent Owner’s Response. Paper 17 (“PO Resp.”). In response, Petitioner filed Petitioner’s Reply to Patent Owner’s Response. Paper 20 (“Pet. Reply”). Patent Owner filed a Sur-Reply to Petitioner’s Reply to Patent Owner’s Response. Paper 22 (“PO Sur-reply”).

The parties requested oral argument (Papers 23, 24), which we held by video on July 13, 2021. Paper 35 (“Tr.”).

The Board has jurisdiction under 35 U.S.C. § 6. In this Final Written Decision, after reviewing all relevant evidence and arguments, we determine Petitioner has met its burden of showing, by a preponderance of the evidence, that claims 1–5 of the ’670 patent are unpatentable.

B. Related Matters

Petitioner and Patent Owner indicate that the ’670 patent was the subject of a patent litigation *BOT M8, LLC v. Sony Corporation of America et al.*, No. 1: 19-cv-07529 (SDNY), which was transferred to the Northern District of California (No. 3:19-cv-07027). Pet. 73; Paper 5, 1. Petitioner

indicates that this case has been dismissed with respect to the '670 patent. Paper 8, 1.

We note a related *inter partes* review pending between the parties, *Sony Interactive Entertainment LLC v. Bot M8, LLC*, IPR2020-01288, that was instituted on February 16, 2021. IPR2020-01288 relates to U.S. Patent No. 7,664,988 (“the '988 patent”), which issued from the parent application (U.S. Serial No. 11/205,121) of the '670 patent. PO Resp. 23.

C. The '670 Patent

The '670 patent discloses “an information process device in which it can be guaranteed that a fault inspection program properly operates even if a fault occurs in a memory device which is inspected through the fault inspection program.” Ex. 1001, 1:36–40. Figure 1 shows an information process device according to an embodiment of the invention.

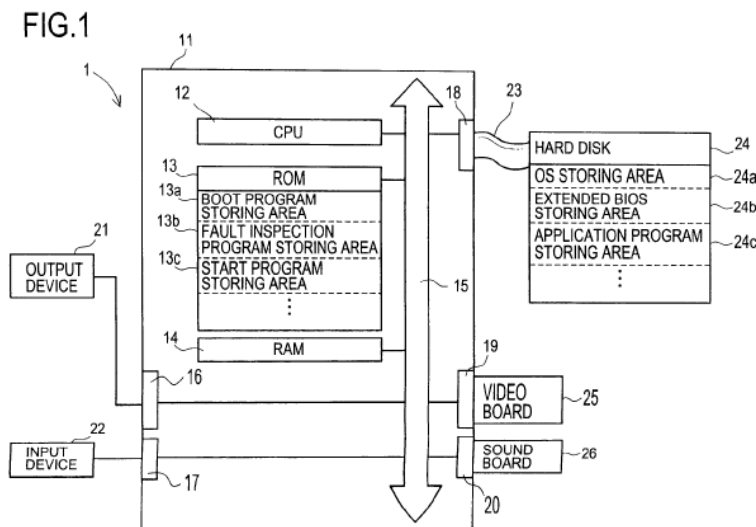


Figure 1 “is a block diagram of an information process device according to the embodiment.” *Id.* at 2:15–16.

As depicted in Figure 1, information process device 1 includes central processing unit (CPU) 12, read-only memory (ROM) 13, and random access memory (RAM) 14 on motherboard 11. *Id.* at 2:36–39. CPU 12 controls

information process device 1 and executes various programs, and therefore, “CPU 12 corresponds to a control device.” *Id.* at 2:40–42. ROM 13 is a non-volatile memory that stores various control programs, including a boot program, a fault inspection program, and a start program. *Id.* at 2:43–51. Hard disk 24 is connected to motherboard 11 at port 18 through cable 23 and stores an operating system (OS), Basic Input Output System (BIOS), and an application program. *Id.* at 1:21–26, 3:19–25.

Information process device 1 begins to operate when CPU 12 executes the start program and, in turn, the boot program. *Id.* at 3:53–58. The boot program initializes the BIOS and the operating system. *Id.* at 3:59–64. The operating system is then loaded into RAM 14 and starts to operate. *Id.* at 3:65–66. Next, the fault inspection program begins to inspect “whether or not a damage occurs in the hard disk 24 or whether or not change or falsification of the program stored in the hard disk 24 is conducted.” *Id.* at 4:1–9. If there is no fault in hard disk 24, the application program is loaded into RAM 14 and begins to execute. *Id.* at 4:15–19. Otherwise, if there is a fault in hard disk 24, an error is displayed on output device 21. *Id.* at 4:19–23. Here, because the fault inspection program is stored in ROM 13, independent from hard disk 24, “even if a fault occurs in the hard disk 24 . . . , it can be guaranteed that the fault inspection program properly operates.” *Id.* at 4:25–31.

D. Illustrative Claims

Petitioner challenges claims 1–5 of the ’670 patent. Pet. 4–72. Independent claims 1 and 4 are illustrative of the challenged claims and are reproduced below:

1. A gaming device configured to execute a game, the gaming device comprising:

a mother board on which a first memory device is provided;

a second memory device configured to store a game application program, the second memory device being connected to the mother board; and

a control device for executing a fault inspection program for the second memory device to inspect whether or not a fault occurs in the second memory device;

wherein the fault inspection program is stored in the first memory device, and the control device completes the execution of the fault inspection program before the game is started.

Ex. 1001, 4:61–5:7.

4. A gaming device configured to execute a game, the gaming device comprising:

a ROM configured to store a fault inspection program;

a memory device which is electrically rewritable a game application program stored therein;

a control device configured to execute the fault inspection program to inspect whether or not a fault occurs in the game application program stored in the memory device;

wherein the control device executes the fault inspection program when the gaming device is started to operate and completes the execution of the fault inspection program before the game is started.

Id. at 5:15–6:10.

E. Instituted Grounds of Unpatentability and Evidence of Record

We instituted the instant trial based on the following grounds of unpatentability.¹ Dec. 6, 35.

Claim(s) Challenged	35 U.S.C. §	References
1–4	103(a)	Sugiyama, ² Gatto ³
5	103(a)	Sugiyama, Gatto, Yamaguchi ⁴
1–4	103(a)	Morrow '952, ⁵ Morrow '771 ⁶

II. ANALYSIS

A. Level of Ordinary Skill in the Art

Petitioner asserts that a person of ordinary skill in the art would have had “the equivalent of at least an undergraduate degree in computer science, computer engineering, electrical engineering, or a similar technical field, and with one or more years of experience in the field of authentication, verification, and/or error detection in the context of computer hardware and/or software.” Pet. 2 (citing Ex. 1003 ¶¶ 49–51). Patent Owner proposes that “a person of ordinary skill in the art [o]n August 17, 2005 (the earliest effective date of the '670 Patent) would have had a Bachelor’s degree in Electrical Engineering, Physics, or a related field, and approximately one or

¹ Petitioner also relies on the Declaration of Andrew Wolfe, Ph.D. Ex. 1003. Patent Owner relies on the Declaration of Long Yang, Ph.D. Ex. 2041.

² JP 2000-35888, published Feb. 2, 2000 (“Sugiyama,” Ex. 1005).

³ WO 2004/004855 A1, published Jan. 15, 2004 (“Gatto,” Ex. 1006).

⁴ US 5,844,776, issued Dec. 1, 1998 (“Yamaguchi,” Ex. 1036).

⁵ US 2004/0054952 A1, published Mar. 18, 2004 (“Morrow '952,” Ex. 1007).

⁶ US 2003/0064771 A1, published Apr. 3, 2003 (“Morrow '771,” Ex. 1008).

more years of professional experience in the field of computer architecture.”
PO Resp. 32 (citing Ex. 2041 ¶¶ 66–70).

We find that both parties propose similar definitions for the level of ordinary skill in the art, with slight variations in the types of professional experience. Specifically, Petitioner proposes a narrower field of experience (“the field of authentication, verification, and/or error detection in the context of computer hardware and/or software”), whereas Patent Owner proposes a broader field of experience (“the field of computer architecture”). Neither party argues that the level of ordinary skill determines the outcome of the patentability analysis. *See* Pet., PO Resp., *passim*.

To the extent necessary herein, we apply Patent Owner’s definition of the level of ordinary skill in the art. We determine the definition offered by Patent Owner is consistent with the teachings of the ’670 patent and the prior art of record. *Cf. Okajima v. Bourdeau*, 261 F.3d 1350, 1355 (Fed. Cir. 2001) (noting that the prior art itself may reflect an appropriate level of skill in the art). We note, however, that neither party explains how the differences in the parties’ competing proposals are material to the issues before us. We further note that our adoption of Patent Owner’s proposed definition does not reflect a view that adopting Petitioner’s competing definition of the level of ordinary skill in the art would have any impact on the outcome of this proceeding. To the contrary, our conclusions would be the same under either party’s definition.

B. Claim Construction

We apply the claim construction standard used to construe the claims in a civil action under 35 U.S.C. § 282(b) articulated in *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc). 37 C.F.R. § 42.100(b) (2019). Under the *Phillips* standard, claim terms must be given “the

meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention.” 415 F.3d at 1313.

1. “*Fault inspection program*”

Independent claims 1 and 4 recite a “fault inspection program.” Petitioner asserts that the ’670 patent explains that a “fault inspection program” is “a program for inspecting whether or not a fault such as damage, change or falsification occurs in the programs or data.” Pet. 20 (citing Ex. 1001, 1:22–25); Reply 6–7. Petitioner argues that “fault” is not limited to mean only damage to the memory device and “change or falsification” to the programs. Reply 6–7. Petitioner emphasizes that the fault inspection program inspects faults in either the hardware or software because the Specification of the ’670 patent does not disclose any requirement to inspect both hardware and software. *Id.* at 7 (citing Ex. 1001, 1:17–18, 4:6–10, claims 1 and 4).

Patent Owner contends that the proper construction of “‘fault inspection program’ is ‘a program, *other than a boot program*, that inspects a memory device for faults, including damage to the memory device and change or falsification of programs stored thereon.’” PO Resp. 23 (citing Ex. 2041 ¶¶ 79–82) (bolding omitted). With this definition, Patent Owner contends that the fault inspection program: (1) is *other* than a boot program, and (2) inspects for faults including damage to the memory device *and* change or falsification of stored programs. *See id.* at 23–28. We address each contention below.

Patent Owner argues that, during prosecution of the ’988 patent, the Applicant distinguished a prior art reference (Bizzarri⁷) “on the basis that

⁷ US 5,732,268, issued Mar. 25, 1998 (“Bizzarri,” Ex. 2001).

that ‘the boot program and the fault inspection program *are distinct*.’” *Id.* at 23 (quoting Ex. 1025, 66) (bolding omitted). Patent Owner argues that this “statement defines the meaning of the terms, and should control their construction.” *Id.* (citing *ProMOS Techs., Inc. v. Samsung Elecs. Co.*, 809 F. App’x 825, 832 (Fed. Cir. 2020) (nonprecedential)); *see also* PO Sur-reply 8 (citing *MBO Labs., Inc. v. Becton, Dickinson & Co.*, 474 F.3d 1323, 1330 (Fed. Cir. 2007)). Patent Owner argues that “fault inspection program” and “boot program” are separately listed claim elements and are thus mutually-exclusive parts of the claimed invention. PO Resp. 24–25 (citing *Becton, Dickinson & Co. v. Tyco Healthcare Grp., LP*, 616 F.3d 1249, 1254 (Fed. Cir. 2010)). Patent Owner further argues that the Specification describes “boot program” and “fault inspection program” as “distinct programs stored in distinct portions of the ROM.” *See id.* at 24, 26–27 (citing Ex. 1001, 2:46–56, 3:59–4:9, Fig. 1).

Petitioner responds that “Patent Owner’s construction improperly reads in numerous unclaimed limitations and is contrary to the intrinsic record.” Pet. Reply 2. Petitioner asserts that Patent Owner’s alleged prosecution history disclaimer during the prosecution of the ’988 patent addressed a claim that recited both a “boot program” and a “fault inspection program,” unlike claim 1 of the ’670 patent. *Id.* at 3–4. Petitioner further asserts that, to the extent the Specification describes embodiments in which “a ‘fault inspection program’ and a ‘boot program’ are stored in distinct portions of the ROM,” the limitations of specific embodiments should not be read into the claims. *Id.* at 4.

We agree with Petitioner. As we determined in our Decision on Institution, the ’670 patent indicates that the “fault inspection program” is “a program for inspecting whether or not a fault such as damage, change or

falsification occurs in the programs or data.” Ex. 1001, 1:22–25; Dec. 17. We see nothing in the record that precludes the “fault inspection program” from also being a “boot program.” Although Patent Owner contends that it presented this argument during the prosecution of the ’988 patent, we are not persuaded to limit the scope of “fault inspection program” as recited in the claims of the ’670 patent. Importantly, Patent Owner does not present any persuasive evidence or argument as to whether the claims of the ’988 patent are substantially similar to those of the ’670 patent, or why Applicant’s statement during the prosecution of a parent to the ’670 patent necessitates a narrowing of the scope of the meaning of the term “fault inspection program” for the claim scope of the ’670 patent. That is, even if we were to consider Applicant’s statement as a disclaimer, we are left to speculate how to construe “fault inspection program” as being distinct from the boot program in the claims of the ’670 patent.

Accordingly, as we did in the Decision on Institution, we construe “fault inspection program” to be “a program for inspecting whether or not a fault such as damage, change or falsification occurs in the programs or data.”

Second, Patent Owner argues that “[a] ‘fault inspection program’ must be capable of inspecting *both* hardware *and* software faults.” PO Sur-Reply 1; PO Resp. 45–47 (bolding omitted). Again, Patent Owner argues that, during prosecution of the ’988 patent, “Applicant reiterated the conjunctive construction of ‘or’ to overcome references such as *Bizzarri*, *Chang*, and *Yamato*.” PO Sur-Reply at 2–3 (citing Ex. 1025, 66, 97); *see* Tr. 45:10 – 46:10. Patent Owner further argues that the Specification “compels a conjunctive interpretation of the word ‘or’” because “the specification confirms that in order to inspect whether a fault occurs in hardware or

software, the fault inspection program must be capable of detecting both types of faults.” *Id.* at 4 (citing PO Resp. 3–4; Ex. 1001, 4:6–9).

Petitioner responds that the Specification lists non-limiting examples of faults, “such as” damage, change, or falsification in programs or data. Pet. Reply 2 (citing Ex. 1001, 1:22–25). Petitioner further asserts that requiring fault inspection of both hardware and software contradicts the Specification’s reference to inspecting a fault in “hardware or software.” *Id.* (citing Ex. 1001, 1:17–18).

We agree with Petitioner. The ’670 patent states that “[t]he present invention relates to an information process device in which a fault in hardware or software is inspected.” Ex. 1001, 1:17–18. We find that a person of ordinary skill in the art would understand from the ’670 disclosure that the faults can be in either hardware or software, or both. We do not agree with Patent Owner that a person of ordinary skill in the art would understand that a description of faults as “software or hardware” uses “or” in the conjunctive sense to mean both software and hardware. *See* Tr. 45:23–46:6.

2. “*Boot program*”

Dependent claim 2 recites a “boot program executed when the gaming device is started to operate.” Petitioner argues that the boot program should be afforded its plain and ordinary meaning as a start-up program that enables a computer to load larger programs. Pet. 4; Reply 6 (citing Ex. 1003, ¶¶ 155, 222–223). Petitioner explains that the Specification does not otherwise limit or diverge from the plain and ordinary meaning. Reply 5. Petitioner relies on two examples provided in the ’670 Specification to support its construction that the boot program may load larger programs (an

extended BIOS program and Operating System). *Id.* (citing Ex. 1001, 3:59–64).

Patent Owner contends that a boot program is “a program that initializes various devices including the extended BIOS and the operating system.” PO Resp. 29 (citing Ex. 2041 ¶¶ 79–81). Patent Owner argues that the Specification supports this interpretation with two descriptions. First, the Specification discloses:

Here, the boot program is a program stored in the boot program storing area 13a of the ROM 13, and based on the boot program, initialization of various devices including the extended BIOS (Basic Input Output System) in the hard disk 24 and the OS (Operating System) in the hard disk 24 is executed.

Id. at 29 (quoting Ex. 1001, 3:57–62) (bolding omitted). Second, Patent Owner argues that the Specification refers to the ROM as “a boot ROM” because it loads and starts the OS. *Id.* (citing Ex. 1001, 3:65–67). Patent Owner further argues that the extrinsic evidence supports a construction that “a ‘boot program’ is a program that is ‘designed to start a computer’ and not simply any program that enables the computer to load a larger one.” PO Sur-reply 9–12 (citing Ex. 2041 ¶ 48; Ex. 1039, 39:20–41:3, 43:13–25).

Petitioner responds that the Specification does not expressly define “boot program.” Pet. Reply 6. Rather, the Specification describes an exemplary embodiment of “a boot program *here* that loads other example programs.” *Id.* at 6 (citing Ex. 1001, 3:59–64). Petitioner distinguishes the Specification’s description from other definitional terminology. *Id.* at 7 (citing *American Med. Sys., Inc. v. Biolitec, Inc.*, 618 F.3d 1354, 1361 (Fed. Cir. 2010)). Petitioner argues that “the example provided in the specification of a boot program that initializes both an extended BIOS and

an OS, is consistent with the broader plain and ordinary meaning of a startup program that loads larger programs.” *Id.* at 8.

We are not persuaded by Patent Owner to limit the scope of the term “boot program” to “a program that initializes various devices including the extended BIOS and the operating system.” Rather, we agree with Petitioner that the term “boot program” should be afforded its plain and ordinary meaning of “a start-up program that enables a computer to load larger programs.” We agree with Petitioner that the example provided in the ’670 Specification is consistent with the plain and ordinary meaning. The example provided in the ’670 patent Specification of a “boot program” is within the context of the ’670 patent and describes the various items that are initialized in the ’670 patent. A person with ordinary skill in the art would understand this example to be consistent with the plain and ordinary meaning of a “boot program” to mean “a start-up program that enables a computer to load larger programs.”

C. Alleged Obviousness of Claims 1–4 over Sugiyama and Gatto

Petitioner contends that claims 1–4 of the ’670 patent are unpatentable under 35 U.S.C. § 103(a) as obvious over Sugiyama and Gatto. Pet. 4–40. As discussed below, Petitioner has demonstrated by a preponderance of the evidence that claims 1–4 would have been obvious over Sugiyama and Gatto. *See* Pet. 4–40.

1. Sugiyama (Ex. 1005)

Sugiyama is directed to “a service program for executing failure diagnosis, restoration, or the like to processing means in a communication terminal.” Ex. 1005, code (57). Specifically, Sugiyama relates to a karaoke device that receives an application program, “for example, a karaoke performance processing program, a singing scoring program, or the like,”

over a communication network, and stores the application program on a hard disk drive. *Id.* ¶ 2.

In an embodiment, a karaoke terminal 3 includes “CPU (processing means and writing means) 20 for controlling portions of the device according to various programs,” “ROM (non-volatile memory) 22,” “RAM 23,” and “hard disk drive (magnetic storage means) 24.” *Id.* ¶ 10. ROM 22 stores “a startup program necessary for starting up the karaoke terminal,” as well as “an initialization program P1 for initializing the hard disk drive 24 and an HDD inspection program P2 for examining the hard disk drive 24.” *Id.* ¶ 11.

When karaoke terminal 3 is turned on, if no abnormality occurs, “the application program stored in the application storage area 24a of the hard disk drive 24 is loaded into the RAM 23, and normal karaoke performance processing is performed.” *Id.* ¶ 22. If an abnormality does occur, “it is determined whether the abnormality is an abnormality relating to the hard disk drive 24.” *Id.* ¶ 23. In such case, “the CPU 20 executes the HDD inspection program P2 stored in the ROM 22.” *Id.* If there is no damage to the hard disk drive itself, for example, “when the stored data is destroyed, or the like, the initialization program P1 is executed by the CPU 20 . . . and initializes the hard disk drive 24.” *Id.* ¶ 24. Backup data may then be written to the hard disk drive. *Id.* If, on the other hand, “restoration is not possible, for example, when the hard disk drive 24 itself is damaged, the manufacturer of the karaoke terminal 3, a restoration company, or the like is requested for repairs.” *Id.*

2. *Gatto (Ex. 1006)*

Gatto is directed to “[a] method for gaming terminals, gaming kiosks and lottery terminals to ensure that the code-signing verification process of downloaded game software can be trusted.” Ex. 1006, code (57). Specifically, when a game operator decides to deploy a new game, a game terminal downloads the code for the game and “executes a program to verify the code signature of the downloaded code.” *Id.* at 10:2–7. “If the downloaded code can be trusted (successfully passes the verification), it is stored locally in persistent memory in the gaming machine.” *Id.* at 10:9–10. When the downloaded code is executed, “the stored signed code is retrieved . . . and its code signature is verified.” *Id.* at 10:14–15. “If the retrieved downloaded code cannot be trusted, the code is trashed or quarantined”; otherwise, “[i]f the retrieved downloaded code can be trusted, it is executed.” *Id.* at 10:15–17. But, because the code-signing verification process “itself might be a fraudulent verification process,” Gatto also “verif[ies] that the code-signing verification platform can be trusted.” *Id.* at 13:11–16.

3. *Analysis of Claim 1*

a) *Petitioner’s Arguments*

The preamble of claim 1 recites a “gaming device configured to execute a game.” Ex. 1001, 4:61. Petitioner argues that, to the extent the preamble is limiting, Sugiyama teaches “a communication terminal’ configured to execute an ‘application program, for example, a karaoke performance processing program, a singing scoring program, or the like,” which a “PHOSITA would have understood . . . is a type of ‘game.’” Pet. 8 (citing Ex. 1005 ¶ 2; Ex. 1003 ¶ 126). Petitioner also argues that, to the

extent Sugiyama's communication terminal is not considered a gaming device, Gatto discloses a "software verification process applicable to download gaming machine software and games, such as for use in a casino." *Id.* at 11 (citing Ex. 1006, 1:3–4, 9:27–28, 10:2–17, Fig. 1; Ex. 1003 ¶¶ 70, 127, 125–130). Petitioner asserts that a person with ordinary skill in the art would have modified Sugiyama's terminal to execute a gaming application program instead of a karaoke application program because "the modification is nothing more than simple substitution of one form of executable program for another, yielding a predictable result (i.e., a communication terminal configured to execute game application programs)." *Id.* at 11–12 (citing Ex. 1003 ¶¶ 125–130).

Claim 1 further recites "a mother board on which a first memory device is provided." Ex. 1001, 4:64. Petitioner contends that the combination of Sugiyama and Gatto discloses this limitation. Pet. 12–17. Petitioner argues that Sugiyama teaches a first memory device, specifically, a ROM that is connected to a bus within a communication terminal. *Id.* at 12–13 (citing Ex. 1005 ¶ 10). Petitioner further contends that Gatto discloses "a motherboard with non-volatile memory components (such as ROM or BIOS) provided thereon and persistent storage media (e.g., HDD or flash memory) connected thereto." *Id.* at 14 (citing Ex. 1006, 4:25, 5:25–26, 6:26, 7:12–17, 13:28–30, 18:25–29, 20:26–30, 22:5–9, 23:19–25; Ex. 1003 ¶¶ 73–76, 131–135). Petitioner argues that a person with ordinary skill in the art would have modified Sugiyama to include the ROM connected to the motherboard because it "amounts to a combination of known elements (i.e., printed circuit boards and common computing components, such as a CPU, RAM, ROM, and HDD) according to known methods (i.e., electrically connecting the components on a main ('mother') board) to obtain a

predictable result—that the components would have worked together in a gaming machine or other computer system, such as a karaoke terminal.” *Id.* at 16 (citing Ex. 1003 ¶¶ 131–135).

Claim 1 also recites “a second memory device configured to store a game application program, the second memory device being connected to the mother board.” Ex. 1001, 4:65–67. Petitioner argues that the combination of Sugiyama and Gatto discloses this limitation. Petitioner argues that Sugiyama teaches a second memory device, specifically a “‘hard disk drive (HDD),’ configured to store a game application program such as a karaoke ‘singing scoring program.’” Pet. 17 (citing Ex. 1005 ¶¶ 2, 10, 12). Petitioner further argues that Gatto discloses connecting a second memory device to a motherboard for the same reasons discussed above. *Id.*

Claim 1 additionally recites “a control device for executing a fault inspection program for the second memory device to inspect whether or not a fault occurs in the second memory device.” Ex. 1001, 5:1–3. Petitioner argues that Sugiyama teaches “a control device, namely the ‘CPU,’ for executing a fault inspection program such as ‘HDD inspection program P2.’” Pet. 20 (citing Ex. 1005 ¶ 23). Petitioner argues that “‘HDD inspection program P2’ inspects whether or not a fault occurs in the second memory device (i.e., ‘hard disk drive (HDD)’).” *Id.* (citing Ex. 1005 ¶ 23). As discussed above, a fault inspection program is “a program for inspecting whether or not a fault such as damage, change or falsification occurs in the programs or data.” *Id.* Petitioner contends that Sugiyama discloses that the CPU executes a process for examining and restoring failures of an area for storing an application program, such as the HDD. *Id.* at 21 (citing Ex. 1005, abstract). Specifically, Petitioner argues Sugiyama discloses that, after an initial startup program, “‘recovery processing’ steps Sa2 and relevant steps

Sa4 through Sa16 are executed by CPU 20.” *Id.* (citing Ex. 1005 ¶¶ 22–28). According to Petitioner, first it is determined whether an abnormality has occurred in the terminal (step Sa2) and whether the abnormality relates to HDD 24 (step Sa4). *Id.* at 22 (citing Ex. 1005 ¶ 22, Fig. 5). That is, Petitioner argues that steps Sa2 and Sa4 through Sa11 are collectively a fault inspection program. *Id.* at 25 (citing Ex. 1005, Fig. 5). Alternatively, Petitioner argues that step Sa5 (“Execute HDD inspection program P2”) is a fault inspection program by itself. *Id.* (citing Ex. 1005, Fig. 5).

Claim 1 further recites “wherein the fault inspection program is stored in the first memory device, and the control device completes the execution of the fault inspection program before the game is started.” Ex. 1001, 5:4–7. Petitioner argues that Sugiyama “teaches that the fault inspection program (e.g., ‘HDD inspection program P2’ alone or in combination with processing steps Sa2 and Sa4 through Sa11 of Fig. 5) is stored in the first memory device (i.e., ROM)” and is separate from the hard disk drive. Pet. 26–28 (citing Ex. 1005 ¶¶ 11, 22, 23, 30, Fig. 3, Fig 5). Petitioner further argues that the “‘HDD inspection program 22’ is executed in step Sa5, which occurs before the game begins in step Sa3.” *Id.* at 28.

Petitioner argues that “[i]t would have been obvious to a PHOSITA to implement the communication terminal of *Sugiyama* as a gaming device instead of (or in addition to) a karaoke device” because “the modification is nothing more than simple substitution of one form of executable program for another, yielding a predictable result.” Pet. 12. In addition, Petitioner argues that “*Gatto* recognizes that communication terminals have been capable of receiving new games and updated software via downloads and executing such downloaded programs since the late 1980s.” *Id.* (citing Ex. 1006, 1:6–8). Alternatively, Petitioner asserts that *Gatto* discloses that

“it is significantly easier to detect fraudulent code prior to its execution than prevent someone to introduce fraudulent code somewhere amongst the gigantic storage disk space, by numerous means, and at unpredictable times.” *Id.* at 31 (quoting Ex. 1006, 29:13–18).

b) Patent Owner’s Arguments

(1) Separate “fault inspection program”

Patent Owner argues that Sugiyama does not disclose a “fault inspection program” separate from a “boot program.” PO Resp. 37. Patent Owner argues that Sugiyama’s steps Sa1–Sa11 are assigned to one of a “Startup program,” “HDD inspection program P2,” or “Initialization program P1,” none of which can be considered the claimed “fault inspection program.” *See id.* at 38–39.

First, Patent Owner argues that Sugiyama’s “HDD inspection program P2” is not a “fault inspection program” because it “does not ‘inspect *whether or not* a fault occurs in’ the second memory device or the game application program.” *Id.* at 39. Patent Owner argues that “HDD Inspection program P2” (step Sa5) “merely ‘checks the details’ of an abnormality that was already previously identified by [the] Startup program.” *Id.* at 39 (citing Ex. 1005 ¶ 23).⁸ Patent Owner argues that because “Startup program” step Sa2 identifies an abnormality, “HDD inspection program P2” does not

⁸ Patent Owner argues that Sugiyama’s “Initialization Program P1,” including steps Sa7–Sa10, is not a fault inspection program. *See* PO Resp. 44. Patent Owner argues that steps Sa7–Sa10 are remediation steps performed “only after an error has been detected and are, therefore, not part of a ‘fault inspection program.’” *Id.* (citing Ex. 1005 ¶ 24, Fig. 5). However, as Petitioner indicates, those steps are “applicable to inspecting the hard disk drive 24.” *See* Pet. 24.

inspect whether or not a fault occurs in the second memory device or the game application program as recited in claim 1. *See id.* at 39–40.

We disagree with Patent Owner. Sugiyama discloses that, in step Sa5, the HDD inspection program P2 is executed. Ex. 1005, Fig. 5. Although an abnormality is detected in steps Sa2 and Sa4, we are not persuaded that step Sa5 does not also execute a program to detect faults. Rather, step Sa5 involves ascertaining the details of the abnormality, which requires identifying the abnormality itself. *Id.* ¶ 23. Accordingly, we find that Sugiyama’s HDD inspection program P2 discloses the claimed “fault inspection program.”

Second, Patent Owner argues that Sugiyama’s “Startup program” is not a “fault inspection program” because it is not separate from a “boot program.” *See* PO Resp. 40–44. Patent Owner argues that the “Startup program” includes at least steps Sa1, Sa2, and Sa4. *Id.* at 40. Patent Owner argues that Petitioner maps step Sa1 to a “boot program.” *See id.* at 40 (citing Pet. 21, 31–33, 35). Patent Owner argues that because steps Sa2 and Sa4 are also part of the “Startup program,” there is no “program distinct and separate from a boot program, which is responsible for detecting abnormalities in the hard disk.” *Id.* at 40. Patent Owner further argues that “Petitioner’s contention that steps Sa2 and Sa4–Sa11 of Figure 5 *collectively* constitute ‘a fault inspection program’ is impossible and overbroad because the Startup program, which performs steps Sa2, Sa4, and Sa11, *cannot simultaneously* be both the ‘boot program’ and the ‘fault inspection program.’” *Id.* at 42 (citing Ex. 1005 ¶¶ 22–23, Figs. 3, 5; Ex. 2045, 79:18–80:7) (bolding omitted).

We are not persuaded by Patent Owner’s argument. First, Patent Owner’s argument is predicated on our adoption of its claim construction for

the term “fault inspection program” as necessarily distinct from a boot program, which we declined to adopt, as discussed above.

See Section II.B.1. Furthermore, as argued by Petitioner, Sugiyama discloses that first it is determined whether an abnormality has occurred in the terminal (step Sa2), and then it is determined whether the abnormality relates to HDD 24 (step Sa4). Pet. 22 (citing Ex. 1005 ¶ 22, Fig. 5). That is, Petitioner argues that steps Sa2 and Sa4 through Sa11 are collectively a fault inspection program. *Id.* at 25 (citing Ex. 1005, Fig. 5). We are not persuaded by Patent Owner that steps Sa2 and Sa4 are limited to being part of the “startup program,” and even if we were persuaded of that point, we disagree that the “startup program” cannot include the “fault inspection program” independent of HDD inspection program P2.

(2) *Detecting hardware and software faults*

Patent Owner argues that “Sugiyama also fails to disclose the claimed ‘fault inspection program’ because it is not capable of determining whether a HDD abnormality is a hardware abnormality . . . or a software abnormality.” PO Resp. 45–46. That is, Patent Owner argues that “Sugiyama’s steps Sa2–Sa11 cannot be a ‘fault inspection program’ because they are not capable of detecting the two types of faults that the inspection program must identify, . . . *i.e.* damage to the memory device and change or falsification of programs stored thereon.” *Id.* at 47. Patent Owner argues that steps Sa2 and Sa4 are typical of a “power-on self-test” that detects memory failures. *Id.* at 46–47 (citing Ex. 2041 ¶ 41).⁹ Patent Owner argues that, after the test, Sugiyama “can load HDD inspection program P2 in order to check the

⁹ PO cites to Exhibit 2014 in the original. However, the Yang Declaration is Exhibit 2041.

details of the abnormality,” but cannot access the file allocation table (“FAT”) of a memory device without an OS loaded. *Id.* at 47 (citing Ex. 1005 ¶ 22; Ex. 2041 ¶ 43).

We disagree with Patent Owner. Patent Owner’s argument is predicated on our adoption of its claim construction, which we decline to do. *See supra* Section II.B.1. Specifically, Patent Owner’s argument requires that we construe “fault inspection program” to be able to detect *both* software *and* hardware faults. However, as discussed above, we are not persuaded that the claimed “fault inspection program” must detect both hardware and software faults. *See id.* Accordingly, we determine that detecting either hardware *or* software faults meets the claim limitation. Patent Owner concedes that Sugiyama discloses abnormalities such as memory failures, which are a type of software fault. *See* PO Resp. 47.

(3) “*Fault inspection program*” *completed before the game is started*

Patent Owner argues that “*Sugiyama* does not disclose a control device that “***completes the execution*** of the fault inspection program ***before the game is started.***” PO Resp. 50. Referring to Sugiyama’s Figure 5, Patent Owner argues that starting the game (step Sa3) following steps Sa1 and Sa2 does not execute HDD inspection program P2. *Id.* at 50–51. Patent Owner argues that starting the game (step Sa3) following process steps Sa2 to Sa4–Sa11 does not involve executing the fault inspection program for the same reasons Patent Owner argues that steps Sa4–Sa11 do not qualify as a fault inspection program. *Id.*

We disagree with Patent Owner. Sugiyama discloses that “if no abnormality occurs in the karaoke terminal 3 (step Sa2), the application stored in the application storage area 24a of the hard disk drive 24 is loaded

into the RAM 23, and normal karaoke performance processing is performed (step Sa3).” Ex. 1005 ¶ 22; *see* Pet. 28–31. That is, the application processing is not started until the “fault inspection program” has completed.

(4) Motivation to combine Sugiyama and Gatto

Patent Owner advances several arguments against Petitioner’s combination of Sugiyama and Gatto. PO Resp. 51–63. The thrust of Patent Owner’s arguments is that Sugiyama’s teachings relate to an embedded OS appropriate for a karaoke machine, in contrast with Gatto’s teaching for adopting a desktop OS for gaming computer hardware. *See id.* at 52–56. Patent Owner argues that modifying Sugiyama’s embedded OS with Gatto’s desktop OS: (1) would render Sugiyama inoperable for its intended purpose (*see id.* at 55–56), (2) would present an obstacle to the combination (*see id.* at 56–58), and (3) would preclude a reasonable expectation of success (*see id.* at 60–63).

Petitioner replies that a “POSITA would have understood that merely connecting Sugiyama’s components to a well-known motherboard, in view of Gatto’s teachings to connect such components to a motherboard, would not have required replacing Sugiyama’s OS with Gatto’s OS.” Pet. Reply 16. Petitioner further argues that the “use of a motherboard, as taught by Gatto, would not render Sugiyama inoperable or change its principle of operation because it would merely be providing a well-known way to connect Sugiyama’s computer components.” *Id.* Petitioner also argues that: (1) Patent Owner does not establish that Sugiyama is limited to an embedded OS (*see id.* at 16–18), (2) obviousness does not require bodily incorporation of the secondary reference into the primary reference (*see id.* at 19), and (3) the claims at issue do not refer to an OS, so that the issues of operating

system implementation are irrelevant to a reasonable expectation of success (*see id.* at 20).

We disagree with Patent Owner. Patent Owner’s argument that “modifying the embedded OS of Sugiyama to accommodate the desktop OS in Gatto would render Sugiyama inoperable for its intended purposes of producing a karaoke machine running a singing scoring program” requires the bodily incorporation of Gatto into Sugiyama. However, bodily incorporation is not the standard for obviousness analysis. *See In re Mouttet*, 686 F.3d 1322, 1332–33 (Fed. Cir. 2012).

Rather, we credit the testimony of Petitioner’s expert, Dr. Wolfe. *See* Pet. 19. Dr. Wolfe explains that a person with ordinary skill in the art would have been motivated to combine different aspects of Sugiyama and Gatto. Ex. 1003 ¶ 123. Dr. Wolfe further explains that Petitioner’s proffered combination of Sugiyama and Gatto is the combination of features known in the art to yield nothing more than predictable results. *Id.* Dr. Wolfe opines that a person with ordinary skill in the art would have been motivated to create “a more versatile system that detects faults in a memory device using a trusted fault inspection program that is guaranteed to operate properly. A PHOSITA would have understood that the use of a motherboard, for example, was common and ordinary in the field.” *Id.* ¶ 124. Dr. Wolfe concludes that a person with ordinary skill in the art “would have been well-equipped with sufficient education, knowledge, and training to make the specific combination . . . with a reasonable expectation of success.” *Id.*

4. *Analysis of Claim 2*

Claim 2 recites the limitations of independent claim 1, and further recites “wherein the first memory device stores a boot program executed when the gaming device is started to operate.” Ex. 1001, 5:8–10. Petitioner argues that Sugiyama teaches a “startup program” provided within ROM 22 (the first memory device). Pet. 31–32 (citing Ex. 1005, Fig. 3, ¶ 11). Petitioner argues that “[t]he startup program loads an application or service program into RAM, and would have been considered by a PHOSITA to be a ‘boot program.’” *Id.* at 32–33 (citing Ex. 1005 ¶¶ 12–13; Ex. 1003 ¶¶ 153–158). Petitioner further argues that “[i]n the context of the service program startup method, the startup program in step Sa1 of Figure 5 loads the service program into RAM.” *Id.* at 33 (citing Ex. 1005 ¶ 13).

Claim 2 also recites “wherein the control device executes the fault inspection program after the boot program is executed.” Ex. 1001, 5:11–12. Petitioner argues that Sugiyama “teaches that CPU 20 (i.e., control device) executes the abnormality inspection program processing for HDD 24 (i.e., ‘HDD inspection program P2’ executed in step Sa5 of Figure 5; or, alternatively, steps Sa2 and Sa4–Sa11 collectively) after the startup program at step Sa1 (i.e., boot program) is executed.” Pet. 35 (citing Ex. 1005, Fig. 5, ¶¶ 22–28; Ex. 1003 ¶¶ 153–158). Petitioner’s arguments as to claim 2 are supported by the cited evidence and are persuasive.

Patent Owner argues that “*Sugiyama’s* alleged fault inspection program is part of its boot program, *i.e.*, the Startup Program.” PO Resp. 63. Patent Owner further argues that “[a]ll of the processing discussed with respect to *Sugiyama’s* FIG. 2, including steps Sa2–Sa11, occurs as part of the boot program and *before* the application program is loaded. Accordingly, *Sugiyama* fails to disclose ‘executes the fault inspection

program *after* the boot program is executed.” *Id.* at 63–64 (bolding omitted). We disagree with Patent Owner. Patent Owner’s argument is substantially the same as presented above with respect to claim 1. Accordingly, we disagree with Patent Owner for the same reasons discussed above. *See supra* Section II.C.3.b.3.

5. *Analysis of Claim 3*

Claim 3 recites the limitations of independent claim 1 and further recites “wherein the second memory device is electrically rewritable.” Ex. 1001, 5:13–14. Petitioner argues that Sugiyama teaches a hard disk device 24 (second memory device) that is electrically rewritable. Pet. 35–36 (citing Ex. 1005 ¶¶ 9, 12, 19, Fig. 4; Ex. 1003 ¶¶ 159–160).

Petitioner’s arguments as to claim 3 are supported by the cited evidence and are persuasive.

Patent Owner does not present separate arguments as to dependent claim 3.

6. *Analysis of Claim 4*

Claim 4 is an independent claim that recites a gaming device similar to claim 1. *See* Ex. 1001, 5:15–6:10. Petitioner presents arguments similar to those previously presented as to claim 1. *See* Pet. 37–44. For example, Petitioner refers to the analysis of claim 1 for some of claim 4’s recited features, including the preamble, “a ROM configured to store a fault inspection program” (see claim 1, first memory device), and “a memory device which is electrically rewritable a game application program stored therein” (see claims 1 and 3, a second, electrically rewritable, memory device). *See id.* at 37–38.

Claim 4 further recites “a control device configured to execute the fault inspection program to inspect whether or not a fault occurs in the game application program stored in the memory device.” Ex. 1001, 6:4–6:7. Referring to the previous arguments as to claim 1, Petitioner argues that “*Sugiyama* teaches a control device, namely the CPU, for executing a fault inspection program such as ‘HDD inspection program P2’ alone or in combination with processing steps Sa2 and Sa4 through Sa11 of Fig. 5.” Pet. 38. Petitioner argues that “it would have been obvious to implement *Sugiyama*’s HDD inspection program P2 to inspect whether or not a fault occurs in a game application program, as taught in *Gatto*.” *Id.* (citing Ex. 1003 ¶¶ 169–172).

Claim 4 also recites “wherein the control device executes the fault inspection program when the gaming device is started to operate and completes the execution of the fault inspection program before the game is started.” Ex. 1001, 6:7–10. Referring to the previous arguments as to claim 1, Petitioner argues that “*Sugiyama* discloses that control device (i.e., the CPU) executes the fault inspection program (i.e., “HDD inspection program P2” at step Sa5 either alone or in combination with processing steps Sa2 and Sa4 through Sa11 of Figure 5) when the gaming device is started to operate.” Pet. 40 (citing Ex. 1003 ¶¶ 173–174). Petitioner argues that “*Sugiyama* further teaches that the CPU completes execution of this program before the game is started, or, alternatively, such an implementation would have been obvious.” *Id.*

Petitioner’s arguments as to claim 4 are supported by the cited evidence and are persuasive.

Patent Owner does not present separate arguments as to claim 4. Instead, Patent Owner argues claims 1 and 4 together. *See* PO Resp. 37–63. We disagree with Patent Owner for the reasons noted above.

D. Alleged Obviousness of Claim 5 over Sugiyama, Gatto, and Yamaguchi

Petitioner contends that claim 5 of the '670 patent is unpatentable under 35 U.S.C. § 103(a) as obvious over Sugiyama, Gatto, and Yamaguchi. Pet. 40–44. As discussed below, Petitioner has demonstrated by a preponderance of the evidence that claim 5 would have been obvious over Sugiyama, Gatto, and Yamaguchi. *See id.*

1. Yamaguchi (Ex. 1036)

Yamaguchi is directed to “[a] static memory device [that] has compatibility with a disk drive installed in an electronic apparatus as an external storage unit.” Ex. 1036, code (57). In an embodiment, “mother board 1 performs a variety of control operations and is provided with a connector 1a,” and “hard disk drive 22 . . . can also be connected to the connector 1a through the flat cable 8.” *Id.* at 6:15–22.

2. Analysis

Claim 5 recites the limitations of independent claim 4, and further recites “wherein the ROM is provided on a mother board having a connector, and the memory device is connectable to the mother board through the connector of the mother board.” Ex. 1001, 6:11–14. Petitioner argues that “Sugiyama in view of Gatto teaches a ROM provided on a motherboard and a hard disk drive (HDD) (i.e., memory device[]) connected thereto.” Pet. 41. Petitioner further argues that, “to the extent disclosure of a specific ‘connector’ is required to satisfy claim 5, the same is disclosed in Yamaguchi.” *Id.* at 41–42. Petitioner contends that Yamaguchi discloses

connector 22a that connects hard disk drive 22 to motherboard 1 via flat cable 8, where the motherboard includes ROM 13. *Id.* at 42 (citing Ex. 1036, 6:6–8, 6:21–23).

Petitioner argues that “it would have been obvious to a PHOSITA that the motherboard and HDD memory device must have been connected to communicate, and a PHOSITA would have been motivated to look to known connectors that provide this necessary connection.” *Id.* at 44. Petitioner argues that Yamaguchi “provides express disclosure of well-known and obvious design choices that would have been easy to implement and, given the common and predictable nature of the electrically connecting computing elements using motherboards at the time of the ’670 Patent, would have had a reasonable expectation of success in doing so.” *Id.*

Petitioner’s arguments as to claim 5 are supported by the cited evidence and are persuasive.

Patent Owner argues that Petitioner has not demonstrated that claim 5 is unpatentable as obvious over Sugiyama, Gatto, and Yamaguchi for the same reasons discussed above with respect to obviousness of claims 1–4 over Sugiyama and Gatto. PO Resp. 64. We disagree with Patent Owner’s arguments for the same reasons discussed above. *See supra* Section II.C.3.b.

E. Alleged Obviousness of claims 1–4 over Morrow ’952 and Morrow ’771

Petitioner contends that claims 1–4 of the ’670 patent are unpatentable under 35 U.S.C. § 103(a) as obvious over Morrow ’952 and Morrow ’771. Pet. 45–72. As discussed below, Petitioner has demonstrated by a preponderance of the evidence that claims 1–4 would have been obvious over Morrow ’952 and Morrow ’771. *See id.*

1. *Morrow '952 (Ex. 1007)*

Morrow '952 is directed to “verifying a device by verifying the components of that device,” for example, “processors, persistent storage media, volatile storage media, random access memories, read-only memories (ROMs), erasable programmable ROMs, data files,” etc. Ex. 1007 ¶ 6. The device may be “a gaming machine, wherein the verification of the gaming machine is performed before game play is allowed.” *Id.* ¶ 19.

In an embodiment, device 10 includes “one or more processors 62, persistent storage media 80 and 90, volatile storage media such as random access memories (RAMs) 76, read-only memories (ROMs) 77 or electrically erasable programmable ROMs (EEPROMs) such as basic input/output systems (BIOS) 64.” *Id.* ¶ 37. Device 10 also includes data files 54, such as software program files 92–96 and operating system files 98. *Id.*

Further, “[e]ither within the device 10, or in the diagnostic system 140 attachable to the device 10, are executable instructions or a software program 70 for verification of the components (verification software 70), which may itself be one of the components 50 to verify if it is internal to the device 10.” *Id.* ¶ 38. “Preferably, the verification software 70 is stored in a basic input/output system (BIOS) 64 device or chip.” *Id.* This “mak[es] it hard to bypass the verification process” “because the code in the BIOS 64 is usually the first code executed upon boot or start-up of the device 10.” *Id.* Thus, “operating system files 98 may be verified before loading or booting, or before any software program 92 is run from the persistent storage media 90. This makes the verification software 70 completely independent of data files 54 stored on the persistent storage media 90 which are being verified.” *Id.* ¶ 64.

2. *Morrow '771 (Ex. 1008)*

Morrow '771 is directed to “gaming machines having the ability to reconfigure entire games, pay tables and/or artwork.” Ex. 1008 ¶ 1. In an embodiment, a gaming platform 70 “enables casino owners to draw off of the large library of casino game functions available in a traditional master processing unit (MPU) stand-alone platform, while adding the graphics and sound capabilities of a personal computer.” *Id.* ¶ 29. Gaming platform 70 includes processor 90 that has a CD-ROM drive for storing “graphics, sound files, presentation software for at least one game, and [a] basic operating system.” *Id.* ¶ 31. Processor 90 also has a “customized BIOS chip, referred to as BIOS+, which provides typical PC boot functions, as well as verification and decryption algorithms.” *Id.* “The gaming platform 70 performs many verification processes during boot-up and game operation,” and “an algorithm that originates on the BIOS+ conducts verification of all files on the CD-ROM.” *Id.* ¶ 37.

3. *Analysis of Claim 1*

a) *Petitioner's Arguments*

The preamble of claim 1 recites a “gaming device configured to execute a game.” Ex. 1001, 4:61. Petitioner argues that, to the extent the preamble is limiting, Morrow '952 teaches “gaming machine 10” configured to execute a game. Pet. 47 (citing Ex. 1007, Fig. 4). Claim 1 further recites “a mother board on which a first memory device is provided.” Ex. 1001, 4:64. Petitioner argues that Morrow '952 teaches “a first memory device, namely ‘electrically erasable programmable ROMs (EEPROMs) such as basic input/output systems (BIOS) 64.’” Pet. 49 (citing Ex. 1007 ¶ 37). Petitioner alternatively argues that “[a] PHOSITA would recognize that

Morrow discloses that the BIOS+ 64 chip is placed on a motherboard,” or “this disclosure would have been obvious to a PHOSITA, especially in light of the explicit teachings in the related Morrow ’771 reference.” *Id.* at 49–50 (citing Ex. 1003 ¶¶ 95–100, 198–204). Petitioner further argues that Morrow ’771 discloses that the “BIOS+ on the Pentium motherboard verifies the CD-ROM before the contents of the CD-ROM can be loaded into the Pentium RAM.” *Id.* at 51 (quoting Ex. 1008 ¶ 31). Petitioner argues that a person with ordinary skill in the art would have been motivated to look to Morrow ’771 “to further understand the commonly described technology, including of the preferred hardware environment for the commonly described BIOS+ chip.” *Id.* Accordingly, Petitioner asserts that the combination of Morrow ’952 and Morrow ’771 “amounts to a combination of known elements . . . according to known methods (i.e., electrically connecting the components on a main (‘mother’) board) to obtain a predictable result.” *Id.* (citing Ex. 1003 ¶¶ 201–204).

Claim 1 also recites “a second memory device configured to store a game application program, the second memory device being connected to the mother board.” Ex. 1001, 4:65–67. Petitioner argues that Morrow ’952 “teaches a second memory device, namely ‘persistent storage media 90,’ configured to store game software application program 92,” and that “[a] PHOSITA would recognize that Morrow discloses that this memory device is connected to the motherboard.” Pet. 52 (citing Ex. 1007 ¶¶ 37–38, Fig. 1; Ex. 1003 ¶ 207). Petitioner further argues that, alternatively, “this disclosure would have been obvious to a PHOSITA, especially in light of the explicit teachings in Morrow ’771.” *Id.* at 52–53. As argued with the preamble, Petitioner contends that Morrow ’771 discloses a Pentium motherboard, and a person with ordinary skill in the art would have connected a second

memory device to the motherboard for the same reasons discussed above.
Id. at 55–56.

Claim 1 additionally recites “a control device for executing a fault inspection program for the second memory device to inspect whether or not a fault occurs in the second memory device.” Ex. 1001, 5:1–3. Petitioner argues that Morrow ’952 teaches “CPU 60 (i.e., control device) for executing verification software program 70 (i.e., fault inspection program) for the persistent storage media 90 (i.e., second memory device) to inspect whether or not a damage, change or falsification (i.e., fault) occurs in the hardware of the media 90 . . . [or] any software programs or data stored on media 90.” Pet. 56. Specifically, Petitioner contends that Morrow ’952 discloses that CPU 60 executes verification program 70 in BIOS+ 64. *Id.* at 57 (citing Ex. 1007 ¶ 38, Fig. 1). According to Petitioner, Morrow ’952 discloses that hardware components are verified by reading an identification number for each hardware component, software is verified by reading a digital signature, and when the read value does not match an expected value, a tilt condition message is generated. *Id.* at 58–60 (citing Ex. 1007 ¶¶ 41, 46, 49, 51; Ex. 1003 ¶¶ 101–111).

Claim 1 further recites “wherein the fault inspection program is stored in the first memory device, and the control device completes the execution of the fault inspection program before the game is started.” Ex. 1001, 5:4–7. Petitioner argues that Morrow ’952 “teaches that the fault inspection program (i.e., ‘verification software 70’) is stored in the first memory device (i.e., ‘BIOS+ 64’),” and “that the control device (i.e., ‘CPU’) completes the execution of this program before the game is started.” Pet. 62–66. Specifically, Petitioner argues that Morrow ’952 discloses that “CPU 60 (control device) completes the execution of verification software 70 before

the game is started, explaining that the verification is done ‘before any software program is run from the storage media.’” *Id.* at 63–64 (citing Ex. 1007 ¶¶ 27, 64).

b) Patent Owner’s Arguments

Patent Owner argues that the combination of Morrow ’952 and Morrow ’771 “fails to disclose the fault inspection program at least because these references are only concerned with verifying programs and do not teach inspecting a memory device for damage.” PO Resp. 64. Patent Owner argues determining that “the identification number is not readable,” as disclosed in Morrow ’952, is not an example of inspecting the media for damage. PO Resp. 65. Patent Owner argues that Petitioner relies on its expert declaration for this feature rather than the express teachings of Morrow ’952. *Id.* (citing Pet. 59; Ex. 1003 ¶¶ 132, 135–137).

We agree with Petitioner. As discussed above, we construe “fault inspection program” to mean “a program for inspecting whether or not a fault such as damage, change or falsification occurs in the programs or data.” *See* Section II.B.1. Petitioner argues that Morrow ’952 discloses that “[v]erification failure may be due to, for example, an error in the data file (i.e., damage) or tampering (i.e., change or falsification) of the data file and/or media 90.” Pet. 61 (citing Ex. 1007 ¶ 52). That is, according to Petitioner, Morrow ’952 determines fault based on damage, change, or falsification. These faults are within the scope of “fault inspection program” under our construction. Accordingly, we are persuaded that Morrow ’952 discloses the verification of hardware or software is by a “fault inspection program.”

Patent Owner further argues that Morrow ’952 “does not state that it does (or can) determine that an error in a data file means that the media is

damaged. Rather, *Morrow* '952 discloses that detecting an error in a data file could indicate either a 'bad sector on a hard disk' or 'that tampering' occurred." PO Resp. 66 (citing Pet. 61) (bolding omitted). Patent Owner's expert analogizes this feature to a "'check engine light' that indicates, *e.g.*, a bad sector had occurred and does not, therefore, make it a 'fault inspection program' as the term is properly construed because it never inspects the media to detect whether the damage is hardware or software damage." *Id.* at 66 (citing Ex. 2041 ¶ 90). Petitioner responds that the proper claim construction "does not require inspecting whether faults occur due to physical damage to hardware. Instead, it encompasses inspecting whether faults occur in programs or data, whether they be due to damage, change, falsification, or any other reason." Pet. Reply 22. Petitioner further argues that "*Morrow* '952's verification program 70 can perform verification to detect errors in data files 54, whether it be, for example, caused by a bad sector on a hard disk (*i.e.*, damage) or whether it be caused by tampering (*i.e.*, change or falsification)." *Id.* at 23 (citing Ex. 1007 ¶ 52).

We agree with Petitioner. As discussed above, we construe the "fault inspection program" as detecting faults in either software or hardware. *See supra* Section II.B.1. *Morrow* '952 discloses verification software to determine whether a data file contains an error. Ex. 1007 ¶ 52. The verification software determines whether the "invalid data file [] is simply an error in the data file [], and not tampering, step 302." *Id.* (emphasis omitted). Accordingly, *Morrow* '952 discloses a "fault inspection program" as claimed because it detects a fault in software.

4. *Analysis of Claim 2*

Claim 2 depends from independent claim 1 and recites “wherein the first memory device stores a boot program executed when the gaming device is started to operate, and wherein the control device executes the fault inspection program after the boot program is executed.” Ex. 1001, 5:8–12.

Petitioner argues that Morrow ’952 teaches

a gaming device according to claim 1 (*see* Claim 1), wherein the BIOS+ 64 (i.e., first memory device) stores a file allocation reader 76 (i.e., boot program) executed when the gaming device is started to operate and that CPU 60 (i.e., control device) executes the verification program 70 (i.e., fault inspection program) only after the file allocation reader 76 is executed.

Pet. 66. Specifically, Petitioner contends that Morrow ’952 discloses that the “file allocation reader 76 is a computer program which comprises a set of executable instructions for processing the file allocation structure such as that used by the operating system 98.” *Id.* at 68 (quoting Ex. 1007 ¶ 63) (underlining omitted). Petitioner argues that Morrow ’952 discloses an operating system that uses a “‘file access system’ that is used at boot up to ‘perform file access operations.’” *Id.* (citing Ex. 1007 ¶ 25; Ex. 1003 ¶¶ 219–224).

Patent Owner argues that “Petitioner’s allegation that Claim 2 is obvious over *Morrow* ’952 in view of *Morrow* ’771 relies entirely on its overbroad construction of the term ‘boot program,’ which contradicts the intrinsic record.” PO Resp. 67 (citing Ex. 1025, 66). Patent Owner argues that Morrow’s “file allocation reader” is executed before booting the operating system. *Id.* at 68. Patent Owner contends that because Morrow’s “file allocation reader” does not initialize the operating system, it does not meet the claimed boot program. *Id.*

We agree with Petitioner. Morrow '952 discloses that processor 60 accesses file allocation reader 76 to open the file allocation structure. Pet. 67–68 (citing Ex. 1007 ¶ 63). This allows the system to verify operating system files before any software program is run. Ex. 1007 ¶ 64. Accordingly, we agree with Petitioner that Morrow '952 discloses execution of the fault inspection program after the boot program is executed, and we disagree with Patent Owner's contrary argument.

5. Analysis of Claim 3

Claim 3 depends from claim 1 and recites “wherein the second memory device is electrically rewritable.” Ex. 1001, 5:13–14. Petitioner argues that Morrow '952 “teaches that persistent storage media 90 (i.e., second memory device) may be a removable flash memory device,” which is electrically rewritable. Pet. 70 (citing Ex. 1007 ¶¶ 13, 52, 59; Ex. 1003 ¶¶ 44, 225–228). Petitioner's argument is supported by the cited evidence and is persuasive.

Patent Owner does not present separate arguments as to dependent claim 3.

6. Analysis of Claim 4

Claim 4 is an independent claim that recites a gaming device similar to claim 1. *See* Ex. 1001, 5:15–6:10. Petitioner presents arguments similar to those previously presented as to claim 1. *See* Pet. 70–72. For example, Petitioner refers to the analysis of claim 1 for some of claim 4's features, including the preamble, “a ROM configured to store a fault inspection program” (see claim 1, first memory device), and “a memory device which is electrically rewritable a game application program stored therein” (see claims 1 and 3, a second, electrically rewritable, memory device). *See id.* at 70–71.

Claim 4 further recites “a control device configured to execute the fault inspection program to inspect whether or not a fault occurs in the game application program stored in the memory device.” Ex. 1001, 6:4–6:7. Referring to the previous arguments as to claim 1, Petitioner argues that Morrow ’952 “discloses CPU 60 (i.e., control device) for executing verification software program 70 (i.e., fault inspection program) for the persistent storage media 90 (i.e., memory device) to inspect whether or not a damage, change or falsification (i.e., fault) occurs in ‘any application programs 92’ stored on media 90.” Pet. 71–72 (citing Ex. 1007 ¶¶ 64; Ex. 1003 ¶¶ 234–235).

Claim 4 also recites “wherein the control device executes the fault inspection program when the gaming device is started to operate and completes the execution of the fault inspection program before the game is started.” Ex. 1001, 6:7–10. Referring to the previous arguments as to claim 1, Petitioner argues that Morrow ’952 “teaches that CPU 60 (i.e., control device) executes verification software 70 (i.e., fault inspection program) upon boot or start-up of the device.” Pet. 72 (citing Ex. 1007 ¶¶ 9, 11, 38). Petitioner also argues that Morrow ’952 “further teaches that the CPU completes execution of this program before the game is started, or, alternatively, such an implementation would have been obvious.” *Id.* (citing Ex. 1003 ¶¶ 236–237).

Petitioner’s arguments as to claim 4 are supported by the cited evidence and are persuasive.

Patent Owner does not present separate arguments as to claim 4. Instead, Patent Owner argues claims 1 and 4 together. *See* PO Resp. 64–67. We disagree with Patent Owner for the reasons noted above.

III. MOTION TO EXCLUDE

The party moving to exclude evidence bears the burden of proving that it is entitled to the relief requested—namely, that the material sought to be excluded is inadmissible under the Federal Rules of Evidence (“FRE”). *See* 37 C.F.R. §§ 42.20(c), 42.62(a).

Patent Owner moves to exclude Exhibits 1040–1044. Paper 27. Patent Owner contends that Exhibits 1040–1044 should be excluded as: (1) hearsay under FRE 801 and 802; (2) not authenticated under FRE 901; and (3) having little probative value under FRE 403. *Id.* at 1. Patent Owner further moves to exclude paragraph 10 of the Reply Declaration of Andrew Wolfe, Ph.D. (Exhibit 1037) because it relies on Exhibits 1040–1044. *Id.* at 4.

Petitioner opposes. Paper 28. Petitioner argues that Exhibits 1040–1044 are not relied on for the truth of the matter asserted therein; rather, Dr. Wolfe cites to these exhibits as “corroborating the knowledge a PHOSITA would have had regarding the embedded nature of the operating systems discussed.” Paper 28, 1–2 (citing Ex. 1037 ¶ 10). Petitioner further argues that these exhibits are admissible because “[t]he ‘appearance, contents, substance, internal patterns, or other distinctive characteristics of the item, taken together with all the circumstances’ are sufficient to authenticate Exhibits 1040–1044 under F.R.E. 901(b)(4).” *Id.* at 4. Finally, Petitioner argues that even if “Exhibits 1040–1044 are themselves inadmissible, Dr. Wolfe may still rely on these exhibits under F.R.E. 703 and his testimony based thereon is not itself inadmissible.” *Id.* at 6. Patent Owner filed a Reply in support of its motion (Paper 29).

We do not rely on Exhibits 1040–1044. We also do not rely on the challenged portion of Exhibit 1037. Accordingly, we dismiss Patent Owner’s Motion to Exclude as moot as to Exhibits 1037 and 1040–1044.

IV. CONCLUSION¹⁰

Based on the information presented, we conclude that Petitioner has shown, by a preponderance of the evidence, that claims 1–5 of the ’670 patent are unpatentable. We dismiss Patent Owner’s Motion to Exclude as moot.

In summary:

Claims	35 U.S.C. §	Reference(s)/Basis	Claims Shown Unpatentable	Claims Not shown Unpatentable
1–4	103(a)	Sugiyama, Gatto	1–4	
5	103(a)	Sugiyama, Gatto, Yamaguchi	5	
1–4	103(a)	Morrow ’952, Morrow ’771	1–5	
Overall Outcome			1–5	

V. ORDER

After due consideration of the record before us, and for the foregoing reasons, it is:

¹⁰ Should Patent Owner wish to pursue amendment of the challenged claims in a reissue or reexamination proceeding subsequent to the issuance of this decision, we draw Patent Owner’s attention to the April 2019 *Notice Regarding Options for Amendments by Patent Owner Through Reissue or Reexamination During a Pending AIA Trial Proceeding*. See 84 Fed. Reg. 16,654 (Apr. 22, 2019). If Patent Owner chooses to file a reissue application or a request for reexamination of the challenged patent, we remind Patent Owner of its continuing obligation to notify the Board of any such related matters in updated mandatory notices. See 37 C.F.R. § 42.8(a)(3), (b)(2).

ORDERED that claims 1–5 of the '670 patent are held unpatentable;
FURTHER ORDERED that Patent Owner's Motion to Exclude is
dismissed as moot;

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.

IPR2020-00726
Patent 8,112,670 B2

FOR PETITIONER:

Eric Buresh
Jason Mudd
Kathleen Fitterling
ERISE IP, P.A.
eric.buresh@eriseip.com
jason.mudd@eriseip.com
kathleen.fitterling@eriseip.com

FOR PATENT OWNER:

James Hannah
Jonathan Caplan
Jeffrey Price
KRAMER LEVIN NAFTALIS & FRANKEL LLP
jhannah@kramerlevin.com
jcaplan@kramerlevin.com
jprice@kramerlevin.com