

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

INTEL CORPORATION,
Petitioner,

v.

BITMICRO, LLC,
Patent Owner.

IPR2023-00787
Patent 10,120,694 B2

Before KEN B. BARRETT, TERRENCE W. McMILLIN, and
KEVIN C. TROCK, *Administrative Patent Judges*.

BARRETT, *Administrative Patent Judge*.

DECISION
Granting Institution of *Inter Partes* Review
35 U.S.C. § 314

I. INTRODUCTION

A. Background and Summary

Intel Corporation (“Petitioner”)¹ filed a Petition requesting *inter partes* review of U.S. Patent No. 10,120,694 B2 (“the ’694 patent,” Ex. 1001). Paper 1 (“Pet.”). The Petition challenges the patentability of claims 1, 5, 6, 9, and 12 of the ’694 patent. BiTMICRO, LLC. (“Patent Owner”)² filed a Preliminary Response to the Petition. Paper 6 (“Prelim. Resp.”).

An *inter partes* review may not be instituted “unless . . . the information presented in the petition . . . shows that there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition.” 35 U.S.C. § 314(a) (2018). Having considered the arguments and evidence presented by Petitioner and Patent Owner, we determine that Petitioner has demonstrated a reasonable likelihood of prevailing on at least one of the challenged claims of the ’694 patent. Accordingly, we institute an *inter partes* review as to all the challenged claims of the ’694 patent on all the grounds of unpatentability set forth in the Petition.

B. Related Proceedings

Both parties identify, as a matter involving or related to the ’694 patent, *BiTMICRO LLC v. Intel Corporation*, No. 5:23-cv-625 (N.D. Cal.). Pet. 71; Paper 4, 2. Petitioner additionally states that the “’694 patent is also

¹ Petitioner identifies Intel Corporation as the real party-in-interest. Pet. 71.

² Patent Owner identifies BitMicro, LLC. as the real party-in-interest. Paper 4, 2.

related to U.S. Pat. No. 9,858,084, which is being challenged in a concurrently-filed petition” in IPR2023-00785. Pet. 72.

C. The '694 Patent

The '694 patent pertains to “computers having a single or multiple Central Processing Units (CPU), and more particularly, to a method and device for booting such a system directly from a storage device.” Ex. 1001, 1:15–18. The patent describes the “use of a minimum amount of non-volatile memory in order to boot an embedded system,” which “brings about an advantage in terms of cost and complexity.” *Id.* at 2:10–14. The disclosed “method also enables the embedded system to initialize all configuration registers even before any of the CPUs of the system is brought out of reset.” *Id.* at 2:23–25.

Figure 2 is reproduced below.

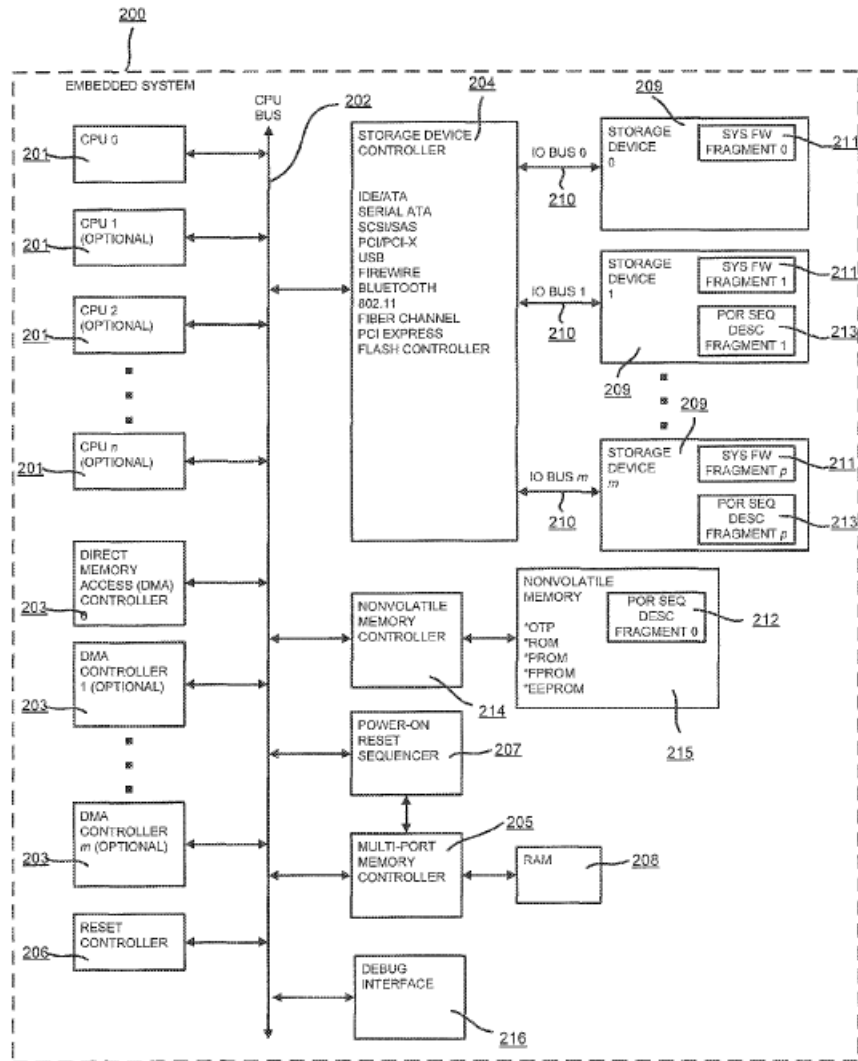
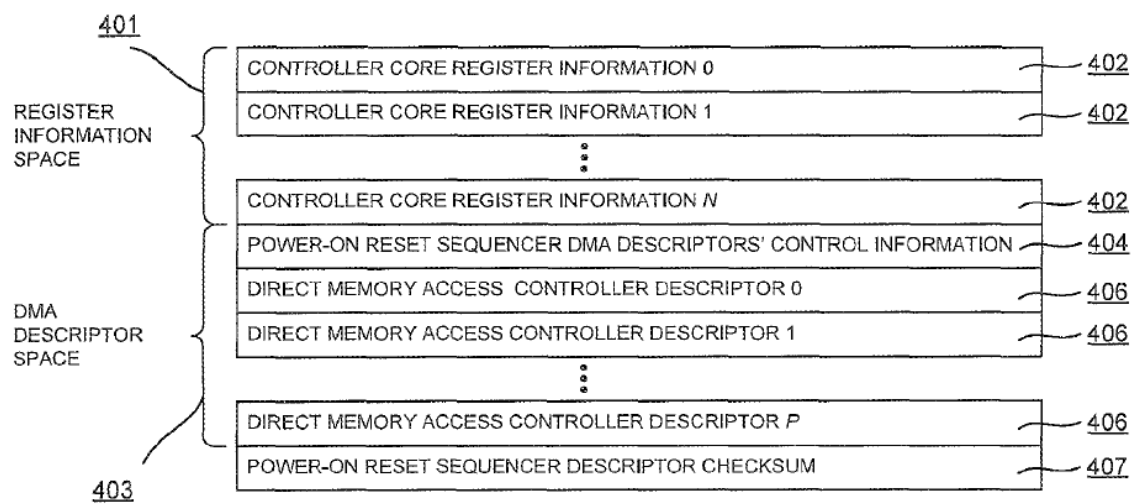


Figure 2 is a block diagram of an embedded system with a power-on reset sequencer that manages the initialization and boot up of the system. Figure 2 shows embedded system 200, including CPU bus 202, CPU 201 (or multiple CPUs 201), direct memory access (DMA) controller 203 (or multiple DMA controllers 203), reset controller 206, power-on reset (POR) sequencer 207, RAM controller 205, nonvolatile memory controller 214, storage device controller 204, debug interface 216, internal RAM 208, small nonvolatile memory 215 and storage device 209. *Id.* at 4:48–56. DMA controller 203 initiates and tracks data transfers. *Id.* at 5:1–2. POR

sequencer 207 processes the POR sequencer descriptor which contains information to initialize configuration registers of the system and DMA descriptors used to fetch other POR sequencer descriptor fragments 213 and the system firmware 211. *Id.* at 5:12–16. POR sequencer 207 also takes care of loading the system firmware to the RAM and verifying its integrity. *Id.* at 5:16–18. POR sequencer 207 copies the POR sequencer descriptor 212 from the nonvolatile memory 215 to the RAM 208, and then checks the integrity of POR sequencer descriptor 212 by comparing the computed checksum to the POR sequencer descriptor checksum 507. *Id.* at 6:64–7:3.

Figure 4, reproduced below, shows the format of the POR sequencer descriptor.



As shown in Figure 4, the '694 patent describes the POR sequencer descriptor as comprising register information space 401, DMA descriptor space 403, and an optional POR sequencer descriptor checksum 407. *Id.* at 6:17–20.

D. Illustrative Claims

Of the challenged claims of the '694 patent, claims 1, 6, and 12 are independent claims. Claims 1 and 6, reproduced below with Petitioner's annotations³ inserted, are illustrative.

1. [1pre] An apparatus, comprising:
 - [1a] an embedded system comprising one or more processors, a reset controller, a storage device controller, one or more direct memory access (DMA) controllers, a random access memory (RAM) and a memory controller, a nonvolatile memory and a nonvolatile memory controller, a debug interface, and
 - [1b] a power-on reset (POR) sequencer configured to detect, in the nonvolatile memory, a descriptor used in a boot process and configured to copy the descriptor from the nonvolatile memory to the RAM,
 - [1c] wherein the POR sequencer facilitates a boot process and verifies an integrity of loaded data and code in the RAM,
 - [1d] wherein the descriptor comprises a POR sequencer descriptor, and wherein the POR sequencer uses the POR sequencer descriptor which is a preassembled descriptor that is stored in the nonvolatile memory,
 - [1e] wherein the POR sequencer descriptor includes register information and DMA controller descriptors, and
 - [1f] wherein the POR sequencer reads and uses the register information to update configuration registers of the embedded system.

6. [6pre] An apparatus, comprising:
 - [6a] an embedded system comprising one or more processors, a reset controller, a storage device controller, one or more direct memory access (DMA) controllers, a random access memory (RAM) and a memory controller, a nonvolatile memory and a nonvolatile memory controller, a debug interface, and

³ Petitioner's annotated versions of the challenged claims are set forth in Appendix B of the declaration of Donald Alpert, Ph.D. (Ex. 1003).

- [6b] a power-on reset (POR) sequencer, wherein the POR sequencer uses a POR sequencer descriptor which is a preassembled descriptor that is stored in the nonvolatile memory,
- [6c] wherein the POR sequencer descriptor includes register information and DMA controller descriptors, and
- [6d] wherein the POR sequencer reads and uses the register information to update configuration registers of the embedded system.

Ex. 1001, 9:13–35, 9:64–10:12.

E. Evidence

Petitioner relies on the following references:

Name	Reference	Exhibit(s)
Kao	US 7,490,177 B2, issued Feb. 10, 2009	1004
Post	US 8,799,555 B2, issued Aug. 5, 2014	1005
Wilson	US 7,861,073 B2, issued Dec. 28, 2010	1013
So	US 7,526,679 B2, issued Apr. 28, 2009	1021

Petitioner also relies on the declaration of Donald Alpert, Ph.D. (Ex. 1003) in support of its arguments. The parties also rely on other exhibits as discussed below.

F. Asserted Grounds of Unpatentability

Petitioner asserts that the challenged claims are unpatentable on the following grounds:

Claim(s) Challenged	35 U.S.C. § ⁴	Reference(s)/Basis
1, 5, 6, 9, 12	103	Post, Wilson, Kao

⁴ The Leahy-Smith America Invents Act (“AIA”) includes revisions to 35 U.S.C. §§ 102 and 103 that became effective on March 16, 2013. Neither party argues, at least at this stage of the proceeding, that the outcome of this case would differ based on applying the pre-AIA or AIA versions of the statutory provisions.

Claim(s) Challenged	35 U.S.C. § ⁴	Reference(s)/Basis
1, 5, 6, 9, 12	103	Post, Wilson, Kao, So

II. ANALYSIS

A. Principles of Law

Petitioner bears the burden of persuasion to prove unpatentability of the claims challenged in the Petition, and that burden never shifts to Patent Owner. *Dynamic Drinkware, LLC v. Nat’l Graphics, Inc.*, 800 F.3d 1375, 1378 (Fed. Cir. 2015).

A patent claim is unpatentable under 35 U.S.C. § 103 if the differences between the claimed subject matter and the prior art are such that the subject matter, as a whole, would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007). The question of obviousness is resolved on the basis of underlying factual determinations including: (1) the scope and content of the prior art; (2) any differences between the claimed subject matter and the prior art; (3) the level of skill in the art; and (4), if present, any objective evidence of obviousness or non-obviousness. *Graham v. John Deere Co.*, 383 U.S. 1, 17–18 (1966).

B. The Level of Ordinary Skill in the Art

In determining the level of ordinary skill in the art, various factors may be considered, including the “type of problems encountered in the art; prior art solutions to those problems; rapidity with which innovations are made; sophistication of the technology; and educational level of active workers in the field.” *In re GPAC Inc.*, 57 F.3d 1573, 1579 (Fed. Cir. 1995)

(quoting *Custom Accessories, Inc. v. Jeffrey–Allan Indus., Inc.*, 807 F.2d 955, 962 (Fed.Cir.1986)).

Petitioner contends that:

A person having ordinary skill in the art (“POSITA”) would possess a Bachelor of Science degree in electrical engineering or computer science, and at least two years of experience in the field of embedded system design, including booting of such systems, or an equivalent combination of education, work and/or experience.

Pet. 17 (citing Ex. 1003 ¶ 20). Patent Owner does not address the level of ordinary skill in the art.

Petitioner’s definition is consistent with the level of ordinary skill reflected in the prior art references of record. *See Okajima v. Bourdeau*, 261 F.3d 1350, 1355 (Fed. Cir. 2001) (recognizing that the prior art itself may reflect an appropriate level of skill in the art). For purposes of this decision, we apply Petitioner’s definition of the person of ordinary skill in the art.

C. Claim Construction

We apply the same claim construction standard used in district court actions under 35 U.S.C. § 282(b), namely that articulated in *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc). *See* 37 C.F.R. § 42.100(b). In applying that standard, claim terms generally are given their ordinary and customary meaning as would have been understood by a person of ordinary skill in the art at the time of the invention and in the context of the entire patent disclosure. *Phillips*, 415 F.3d at 1312–13. “In determining the meaning of the disputed claim limitation, we look principally to the intrinsic evidence of record, examining the claim language itself, the written description, and the prosecution history, if in evidence.” *DePuy Spine, Inc.*

v. Medtronic Sofamor Danek, Inc., 469 F.3d 1005, 1014 (Fed. Cir. 2006) (citing *Phillips*, 415 F.3d at 1312–17).

Petitioner states that it “does not believe any term requires construction to resolve the patentability disputes in this proceeding,” and that “[t]he Petition applies the plain and ordinary meaning to the language of the challenged claims.” Pet. 17.

Patent Owner argues that “Petitioner’s proposed characterization of the claimed ‘DMA controller descriptors’ as satisfied by Post’s pointer is incorrect, and does not comport with the plain and ordinary meaning of this claim term.” Prelim. Resp. 10. Patent Owner further argues that “this characterization fails to consider the ’694 Patent’s specification disclosure that the ‘DMA controller descriptors’ allow for activation of associated DMA controllers.” *Id.* Patent Owner “submits that the Board need not engage in any pre-institution claim construction to deny the Petition.” *Id.* at 10–11.

Thus, neither party contends that any claim construction is necessary at this time. On this record and for purposes of this decision, we determine that no claim terms require express construction. We address below, in analyzing Petitioner’s application of the prior art to the claim limitations, Patent Owner’s arguments regarding “direct memory access (DMA) descriptors” as recited in the challenged claims.

D. The Alleged Obviousness of Claims 1, 5, 6, 9, and 12 over Post, Wilson, and Kao (Ground 1) and over Post, Wilson, Kao, and So (Ground 2)

Petitioner alleges that claims 1, 5, 6, 9, and 12 of the ’694 patent would have been obvious over Post, Wilson, and Kao and over Post, Wilson, Kao, and So. *See* Pet. 27–66 (addressing independent claim 1 under both

grounds). Of those challenged claims, claims 1, 6, and 12 are independent.

Petitioner contends, *inter alia*, that:

In the resulting combination of Post, Kao, and Wilson, Post's control circuitry 112 would be modified to include Wilson's register file and circuitry to perform the register loading functionality of Wilson's RLA engine as well as a DMA controller that performed the boot data transfer functionality of Kao's DMA engine; and Post's boot data included initial register values, as taught by Wilson. In the resulting combination of Post, Kao, Wilson, and So, Post's SoC [system-on-chip] 110 would be further modified to include So's reset controller.

Pet. 27 (citing Ex. 1003 ¶ 73).

1. Post (Ex. 1005)

Post is titled "Boot Data Storage Schedules for Electronic Devices." Ex. 1005, code (54). Post discloses systems and methods "for storing and retrieving boot data (e.g., a first stage bootloader) in and from a nonvolatile memory ('NVM'), such as a NAND flash memory." *Id.*, Abstr. Post describes that to "prevent a boot ROM [Read Only Memory] from having to maintain a NVM-specific map of which pages are used to store boot data, the map may be maintained in the NVM itself," such as "in the form of a linked list, where each page storing boot data can include a pointer that points to the next page that stores boot data." *Id.*

Post's Figure 1 is reproduced below.

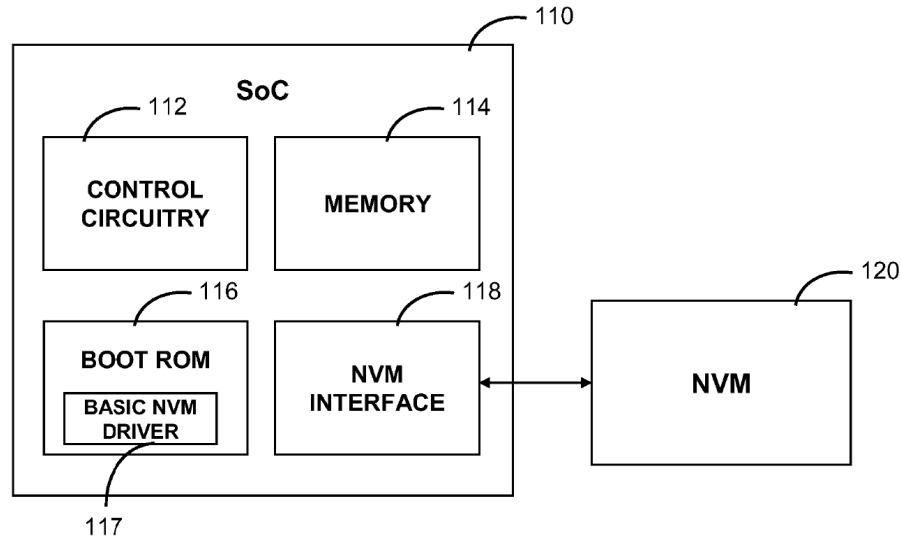


Figure 1 is a block diagram of an electronic device with non-volatile memory. Ex. 1005, 2:36–38. Figure 1 shows electronic device 100, which may be a portable media player, cellular telephone, pocket-sized personal computer, personal digital assistant, desktop or laptop computer, or other device. *Id.* at 2:60–67. The device includes system-on-a-chip 110, control circuitry 112, memory 114, boot ROM 116, NVM interface 118, and non-volatile memory 120. *Id.* at 3:33–36. Boot data may be stored in a subset of memory locations in the boot data storage area of NVM 120. *Id.* at 4:42–47. The subset of pages selected to store boot data may be nonconsecutive. *Id.* at 5:39–40. According to Post, “the loading of boot data can be accomplished by incrementally processing and validating pages as the pages are being read.” *Id.* at 5:50–52.

Post describes that each block of data in NVM can store both boot data and metadata, where the metadata can include a “next page pointer” and “prior page pointer, and “boot data page indicator.” *Id.* at 6:28–37. Post discloses that:

the electronic device can use the next page pointers to verify the integrity of boot data storage. For example, since next page

pointer 302 points to page 2, page 2 is expected to contain boot data. The electronic device may confirm that page 2 provides boot data by, for instance, employing an error correcting code on the contents of page 2.

Id. at 6:54–59.

2. *Wilson (Ex. 1013)*

Wilson relates “to microcontrollers comprising on-chip decompression engines.” Ex. 1013, 1:5–7. Wilson explains that a “microcontroller is often referred to as ‘a computer on a chip.’” *Id.* at 1:13–14.

Wilson explains that “[m]any microcontrollers require a boot-up process when initially powered up,” which “allows a microcontroller to perform various actions to prepare the microcontroller for ordinary operation, such as load initialization values in its registers, load instructions in its instruction memory, etc.” *Id.* at 1:24–28. Wilson further explains that “removing errors in a microcontroller’s boot operation becomes difficult if all boot data of a microcontroller is stored internally in a ROM,” but “storing the boot data outside of the microcontroller poses other difficulties.” *Id.* at 1:40–45.

To address this, Wilson discloses a “microcontroller which includes a Register Load Assist (RLA) engine.” *Id.* at 1:55–57. In this way, “most of the boot data can be stored at a non-volatile memory external to the microcontroller.” *Id.* at 1:58–60. “The RLA engine can then decompress the boot data and store various portions of it in their final destinations (such as, for example, in respective registers).” *Id.* at 2:1–3. As a result, “by allowing the remote receipt of compressed boot data, the RLA engine provides for a microcontroller which can receive a large amount of boot data

from an external source, but nevertheless feature a relatively fast boot-up sequence.” *Id.* at 2:4–8. Figure 1 of Wilson is reproduced below.

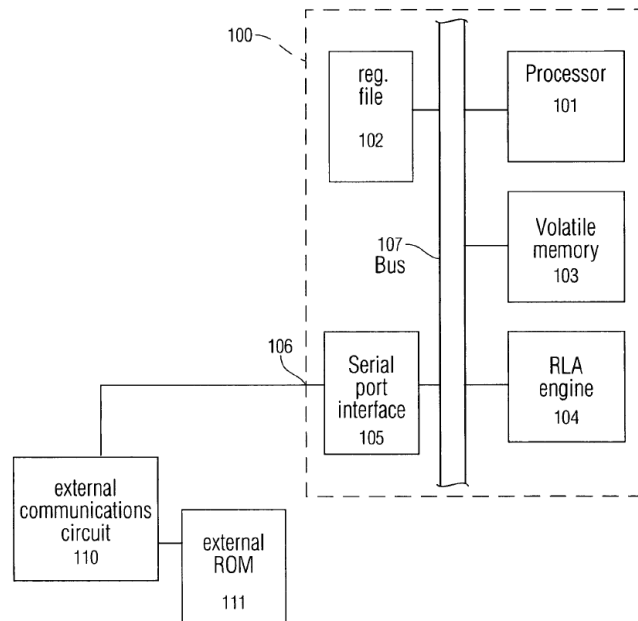


Figure 1 is a diagram of Wilson’s microcontroller and related components. *Id.* at 2:12–14. Figure 1 shows microcontroller 100, programmable processor 101, register file 102, volatile memory 103, RLA engine 104, tied together with bus 107, as well as external ROM 111, which may store boot and other data, and which can send the boot data to the microcontroller through port 106 and serial port interface 105. *Id.* at 2:51–3:32.

3. *Kao (Ex. 1004)*

Kao “relates to an improved memory architecture for booting a computer system.” Ex. 1004, 1:10–13.

Kao discloses that devices “such as cell phones, PDAs, portable music players, appliances, and so on typically incorporate an embedded computer system.” *Id.* at 1:15–17. Kao describes when an “embedded system is powered up, the embedded system typically performs a boot sequence to properly load the operating code from the nonvolatile memory to the volatile

memory and initialize the host with the operating code.” *Id.* at 1:43–46. “In order to perform the boot sequence, the host typically accesses boot code which is stored in a predefined area of non-volatile memory.” *Id.* at 1:46–49. Kao further explains:

In some cases, the host may transfer the operating code from the nonvolatile memory to the volatile memory using direct memory access (DMA). DMA allows a portion of memory to be quickly and automatically moved from one storage location to another storage location. To perform the DMA transfer, the host may need to initialize the DMA engine. In some cases, the DMA engine may be located in the host. To initialize the DMA engine, the host may use information and/or instructions provided by the boot code. Once the DMA engine is initialized, the host may issue commands to the DMA engine to load operating code from the non-volatile memory into the volatile memory. The host may then begin executing the operating code and any other code necessary for proper operation of the host.

Id. at 1:54–67.

Kao discloses that:

By utilizing the DMA engine located on the volatile memory chip to load boot and controller codes, the processing workload necessary to boot the embedded system may be offloaded from a host processor to the volatile memory chip, thereby reducing the complexity (and therefore the overall cost) of the host processor and the embedded system.

Id. at 3:3–9. Then, the “host may perform boot operations,” which “may include issuing commands to the DMA engine to load operating system code in the volatile memory.” *Id.* at 3:13–17.

4. *So (Ex. 1021)*

So relates to “an apparatus for developing and verifying a system-on-chip for [an] Internet phone integrating a core module, a field programmable

gate array (FPGA), a peripheral interface and the system-on-chip.”

Ex. 1021, 1:7–12.

So discloses that its apparatus “includes an Advanced RISC Machine (ARM) core module 11, a field programmable gate array (FPGA) 12, a peripheral interface 13, and a system-on-chip 14.” *Id.* at 2:58–62. So’s ARM code module 11 “includes an ARM core 21 having a cache, a Multi-ICE synchronizing block 22, a clock generating block 23, an instruction tightly coupled memory (ITCM) 24, a data tightly coupled memory (DTCM) 25, an embedded trace module (ETM) 26 and an advanced high-performance bus (AHB) coupling block 27.” *Id.* at 3:26–32. So’s “system-on-chip 14 includes a core processor block 310 performing a function of a central processing unit, a user input/output block 330 for connection with diverse external input/output devices, and a network accessing block 340 for connection with the external network.” *Id.* at 3:56–61.

So further discloses that within the system-on-chip 14:

The core processor block 310 includes a core processor 311 connected to the AHB, a reset controller 312, a clock controller 313, an external bus controller 314, an AHB decoder 315, a bus bridge 316, an interrupt controller 317, a direct memory access controller (DMAC) 318, an internal static random access memory (SARM) 319, an advanced peripheral bus (APB) decoder 320 connected to the APB, a real-time clock (RTC) 321, a watch dog timer (WDT) 322 and a dual timer 323.

Id. at 3:62–4:3.

According to So, the “reset controller 312 connected to the external reset block initializes all controllers according to the reset signal inputted

from the external reset block, generates the reset signal required in each controller and provides the signal to the AHB.” *Id.* at 4:4–8.

Figure 3 of So, reproduced below, is a block diagram of the system-on-chip 14.

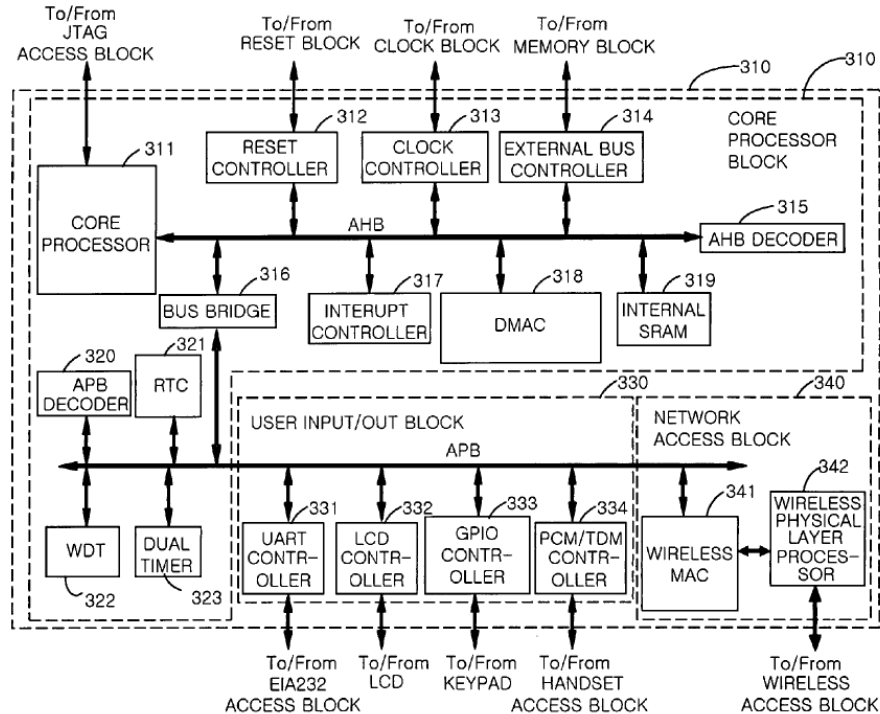


Figure 3 shows So’s system-on-chip 14, including core processor block 310, which includes core processor 311 and reset controller 312. *Id.* at 3:62–4:3.

5. *The Alleged Obviousness of Independent Claims 1 and 12 and Dependent Claims 5 and 9 over Post, Wilson, and Kao (Ground 1) and over Post, Wilson, Kao, and So (Ground 2)*

For the reasons discussed below, Petitioner has not shown a reasonable likelihood that it would prevail in establishing unpatentability under either ground of any of independent claims 1 and 12 and dependent claims 5 and 9.

Independent apparatus claim 1 recites “wherein the POR sequencer . . . verifies an integrity of loaded data and code *in the RAM*.” Ex. 1001, 9:24–26 (element 1c) (emphasis added). Claim 5 depends directly

from independent claim 1 and therefore also contains the same requirement. *See id.* at 9:62–63. Independent claim 12, directed to a computer readable medium having instructions, contains a similar recitation. *Id.* at 10:53–55 (“verifying, by the POR sequencer, an integrity of loaded data and code in the RAM”). Claim 9, which depends directly from independent apparatus claim 6, the only other remaining challenged independent claim, similarly recites “wherein the POR sequencer . . . verifies an integrity of loaded data and code in the RAM.”⁵ *Id.* at 10:35–37. Thus, all of these claims recite an integrity verification of data “in the RAM.”

Patent Owner argues that “the claimed requirement is to verify the integrity of the data and code *loaded in the RAM*, not boot data still stored in non-volatile memory.” Prelim. Resp. 16. On the current record, we agree. Patent Owner’s argument is consistent with the claim language—referring to data and code “in the RAM”—and is consistent with, for example, the Specification’s Figure 6 depicting a flow chart having integrity verification step 606 near the end of the process and after step 605, where the POR sequencer copies the POR sequencer descriptor from ROM to RAM. *See* Ex. 1001, Fig. 6; *see also id.* at 6:64–7:3 (describing Figure 6).

Patent Owner argues that “[t]he Petition fails to establish that these features are present or suggested by Petitioner’s proposed combination of Post, Kao, and Wilson” or the combination of Ground 2 further including So. Prelim. Resp. 15; *id.* at 22 (“Each of the shortcomings identified above

⁵ Although Patent Owner asserts that, “[f]or the purpose of Petitioner’s challenges, claims 6 and 12 are substantially similar to claim 1” (Prelim. Resp. 7), independent claim 6 does not contain the “verifies an integrity” phrase. *See* Ex. 1001, 9:64–10:12. Rather, such a limitation is added to the claimed subject matter by dependent claim 9.

also apply to Petitioner’s challenge based on the asserted combination of Post, Kao, Wilson, and So.”). Specifically, Patent Owner argues that the portions of Post upon which Petitioner relies discloses verifying and accounting for all the boot data stored *in the non-volatile memory* (NVM), not data that has been loaded into RAM. *See id.* at 15–17.

Petitioner contends that “[c]ontrol circuitry 112 verified an integrity of the boot data that was loaded into memory 114 (*‘loaded data and code in the RAM’*) by using one or more of next page pointers 302, prior page pointers 304, and boot data page indicators 306.” Pet. 55; *see also id.* at 69 (for dependent claim 9, relying on the same arguments), 71 (for independent claim 12, relying on the same arguments). In support, Petitioner quotes Post’s statement: “[T]he electronic device can use the next page pointers to verify the integrity of boot data storage.” *Id.* at 55 (quoting Ex. 1005, 6:54–55; citing 6:22–24, 7:21–23, 7:58–64, 8:52–66, 9:33–46, 10:52–54; Ex. 1003 ¶¶ 138–139). The quoted statement and some of the references to pointers are from Post’s discussion of Figure 3, which is “a graphical view of two illustrative blocks . . . of a *non-volatile memory*.” Ex. 1005, 6:21–22 (emphasis added), 6:28–37. In another of Petitioner’s cited provisions, Post states that, “[a]s with the next page pointers, the prior page pointers may allow an electronic device to verify and account for all of the boot data stored in the NVM.” *Id.* at 7:21–23. Post explains that pointers can be used “to verify and account for all of the boot data stored in the NVM.” *Id.* at 7:29–33; *see also, e.g., id.* at 9:41–44 (“[T]he electronic device can perform several checks on the validity and accuracy of the boot data stored in the non-volatile memory.”). However, as Patent Owner argues, the claims recite verifying information “in the RAM,” not in the non-volatile memory.

Petitioner further contends that “Post also disclosed that control circuitry 112 used error correcting codes to verify an integrity of the boot data that was loaded in memory 114,” and that a person of ordinary skill in the art “would have understood that these functions were performed by control circuitry 112, and would also find this obvious, because Post described control circuitry 112 as verifying the integrity of boot data that was loaded from nonvolatile memory using pointers and error correcting codes.” Pet. 55–56 (citing Ex. 1005, 3:34–36, 6:57–59, 8:9–12, 9:25–46; Ex. 1003 ¶¶ 140–141). Patent Owner, in response, argues that “according to Post, the use of error correcting codes is on data that has not been loaded out of nonvolatile memory.” Prelim. Resp. 17 (citing Ex. 1005, 6:21–24, 6:54–59). On the current record, we find Patent Owner’s argument persuasive.

After considering the cited portions of Post and of Dr. Alpert’s testimony, we determine that Petitioner has failed to explain adequately how the relied on prior art discloses or suggests a POR sequencer that “verifies an integrity of loaded data and code in the RAM,” as recited in several of the challenged claims. Accordingly, Petitioner has not demonstrated a reasonable likelihood of prevailing on its challenges to independent claims 1 and 12 and dependent claims 5 and 9.

6. The Alleged Obviousness of Independent Claim 6 over Post, Wilson, and Kao (Ground 1) and over Post, Wilson, Kao, and So (Ground 2)

As mentioned above, Petitioner contends that independent claim 1 would have been obvious over Post, Wilson, and Kao, and also those same references further in combination with So. Pet. 27–66. For independent claim 6, Petitioner relies on the same contentions. *See id.* at 69 (including a summary table cross-referencing contentions on an element-by-element

basis). Petitioner contends that Post discloses much of the claimed subject matter (including a reset controller), relies on Kao for teachings related to a DMA engine, on Wilson for teachings related to register values, and, alternatively, on So for teachings related to a reset controller. *See id.* at 27 (describing the proposed modifications); *see also id.* at 29–32 (addressing the reset controller).

Patent Owner argues that “Petitioner’s combination [of Post with Kao] is not supported by any evidence of record, and the resulting combination fails to disclose a POR sequencer performing the claimed functions as recited in claim 1.” Prelim. Resp. 11, 15. Patent Owner also argues that the proposed combination fails to disclose or suggest a POR sequencer descriptor that includes DMA controller descriptors. *Id.* at 18–21. Patent Owner asserts that “the remaining challenged claims [including independent claim 6] . . . recite features substantially similar to claim 1,” and, for claim 6, relies on the same arguments made for claim 1. *See id.* at 21–22. Patent Owner does not present separate arguments for Ground 2, which includes So. *See id.* at 22 (“Petitioner’s addition of So to the Ground 1 combination of Post, Kao, and Wilson is provided to allegedly disclose the claimed ‘reset controller.’ However, this addition of So is not provided to remedy the above-identified shortcomings of the Post-Kao-Wilson combination.”).

For the reasons discussed below, Petitioner has shown a reasonable likelihood that it would prevail in establishing unpatentability of independent claim 6 under both grounds.

a. [6pre] An apparatus, comprising:

Petitioner contends, to the extent the preamble is limiting, that:

Post disclosed an “electronic device 100” (an “apparatus”) such as “a cellular telephone (e.g., an iPhone[]),” “a desktop computer, a laptop computer, and any other suitable type of electronic device or system.” (INTEL-1005, 2:60-67.) Device 100 included a system-on-chip 110 (*id.*, 3:1-2), which included control circuitry 112 that “control[led] the general operations and functions of SoC 110 and electronic device 100” and “include[d] any suitable components, circuitry, or logic, such as one or more processors.” (INTEL-1005, 3:33-38; INTEL-1003, ¶ 75.)

Pet. 27.

Patent Owner does not raise any arguments regarding these contentions.

Petitioner has demonstrated sufficiently, for purposes of this Decision, that Post discloses the preamble.

b. [6a] an embedded system comprising one or more processors, a reset controller, a storage device controller, one or more direct memory access (DMA) controllers, a random access memory (RAM) and a memory controller, a nonvolatile memory and a nonvolatile memory controller, a debug interface, and

Petitioner contends that “Post’s system-on-chip (‘SoC’) 110 and nonvolatile memory 120 were collectively *an embedded system*.” Pet. 28 (citing Ex. 1003 ¶¶ 76–77; Ex. 1005, 3:33–38 (“Control circuitry 112 [of SoC 110] can include any suitable components, circuitry, or logic, such as one or more processors.”)).

For the recited “reset controller,” Petitioner contends that “Post described a process for booting up an electronic device in response to a ‘startup or reboot command’ received by control circuitry 112,” and that a person of ordinary skill in the art “would have understood Post’s disclosure of control circuitry 112 receiving a ‘startup or reboot command’ to indicate

the presence of a reset controller in SoC 110 that generated these commands.” *Id.* at 29–30 (citing Ex. 1003, ¶¶ 82–83; Ex. 1005, 1:16–18, 8:9–12, 8:15–27, 9:7–11; Ex. 1016, 555 (“System boot-up means that some type of power-on or reset source . . . has occurred.”)).

Petitioner further contends that:

A [person of ordinary skill in the art] would have understood power-on reset to be one of “[t]he most common microprocess-supervisory functions” (INTEL-1018, 1) and, at a minimum, would have viewed Post’s disclosure of control circuitry 112 receiving startup or reboot commands as providing a suggestion to incorporate a conventional reset controller into SoC 110. (INTEL-1003, ¶ 83; INTEL-1008, 340 (illustrating an integrated reset controller).)

Id. at 30.

Petitioner, in the alternative, contends that So “disclosed a reset controller 312 within a core processor block of a system-on-chip for an Internet phone (an ‘embedded system’),” and that it would have been obvious to modify Post such that the SoC included So’s reset controller 312. Pet. 30–31 (citing Ex. 1021, 3:53–55, 3:62–63, Abstract). Petitioner, relying on the testimony of Dr. Alpert, reasons that:

So described a system on chip (SoC) for inclusion in a device like a phone, similar to Post’s SoC 110 (which was included in, e.g., an iPhone). So taught a reset controller that was included in the SoC and that generated reset signals for different controllers in the SoC according to an external reset signal. Because Post described control circuitry 112 as receiving this type of signal at boot up, but was silent about how it was generated, a POSITA [(person of ordinary skill in the art)] would have looked to So for implementation details.

Additionally, a POSITA would have understood this combination to be obvious because So taught a known technique for solving a known problem (a reset controller for providing reset signals for initializing embedded system

components); and incorporating a reset controller into Post's SoC 110 would have been within the skill of an ordinary artisan, who would be generally familiar with reset controllers, and because So showed that a POSITA knew how to do so for a SoC structurally similar to Post's SoC.

Id. at 31–32 (citations omitted).

As to the recited “storage device controller,” Petitioner contends, *inter alia*, that a person of ordinary skill in the art “would have understood Post's disclosure that NVM [(non-volatile memory)] interface 118 included ‘circuitry or components’ that ‘enable[d] SoC 110 to access’ the memory locations of nonvolatile memory 120's distinct circuits to indicate that NVM interface 118 included a ‘storage device controller.’” *Id.* at 33 (citing Ex. 1003 ¶ 88; Ex. 1001, 2:40–41) (second alteration in original).

As to the recited “random access memory (RAM) and a memory controller,” Petitioner first contends that “Post disclosed that SoC 110 included memory 114 (INTEL-1005, 3:33–34), which was ‘a random access memory (RAM).’” *Id.* at 38. Petitioner next contends that “Post disclosed that memory 114 acted as ‘the main memory for any processors implemented as part of control circuitry 112’ and that control circuitry 112 wrote data to, and read data from, memory 114,” and that, “[g]iven this disclosed interoperability between control circuitry 112 and memory 114, a POSITA would have understood that SoC 110 included a memory controller, and at a minimum, Post suggested including a memory controller in SoC 110, to enable control circuitry 112 to interface with memory 114 in the manner described.” *Id.* (citing Ex. 1005, 3:50–51, 3:55–57, 4:5–8; Ex. 1003 ¶¶ 95–96; Ex. 1009, 110).

As to the recited “nonvolatile memory and a nonvolatile memory controller,” Petitioner contends that “Post disclosed nonvolatile

memory 120” and “that nonvolatile memory 120 included a ‘memory controller’ such as a ‘flash controller.’” *Id.* at 38–39 (citing Ex. 1005, 3:1–2, 4:19–23, Fig. 1). Petitioner further contends that a person of ordinary skill in the art “would have understood Post’s disclosure of a flash controller to be a nonvolatile memory controller, and a POSITA would have understood Post’s disclosure of a flash controller that performed the functionality of a nonvolatile memory driver and was included in nonvolatile memory 120 to be an example of a ‘nonvolatile memory controller.’” *Id.* at 39 (citation omitted).

As to the recited “debug interface,” Petitioner asserts that “[t]he ’694 patent describes a ‘debug interface’ as including, for example, a component ‘used to preprogram the nonvolatile memory and storage device with data and code,’” and Petitioner describes what it asserts to be Post’s disclosures along these lines. *Id.* at 39–40 (quoting, *inter alia*, Ex. 1001, 2:43–45). Petitioner contends that, “[t]hus, SoC 110 included a ‘debug interface,’” and,

[a]t the very least, a POSITA would have understood Post’s disclosure of “changing or updating” stored boot data to suggest the inclusion of a debug interface (like a JTAG port) in Post’s SoC 110 because these components were commonly included in processors for enabling “real-time debugging of hardware” (INTEL-1008, 157-59) and because a POSITA would have understood this to provide a means of implementing the “changing or updating” boot data functionality.

Id. at 40 (citing Ex. 1003 ¶ 102).

Patent Owner does not raise any arguments regarding these contentions.

Petitioner turns to Kao for the recited “one or more direct memory access (DMA) controllers,” contending that “Kao disclosed a ‘DMA

engine 120' within embedded system 100," that "Kao taught that 'the source address and size of the boot code [to be loaded] may be provided' to DMA engine circuitry 120 by, for example, an external device," and that "DMA engine circuitry 120 then automatically loaded boot code from nonvolatile NAND flash memory 130 to a RAM memory device 110." *Id.* at 33–34 (citing Ex. 1004, 1:17–24, 3:22–37, 3:46–51, 3:61–65, 4:20–26, 5:5–7, 5:12–14, Fig. 1). According to Petitioner, "Kao's DMA engine circuitry 120 was thus a 'DMA controller.'" *Id.* at 34 (citing Ex. 1003 ¶¶ 89–90).

With regard to combining the relevant teachings of Post and Kao, Petitioner contends that:

A POSITA would have found it obvious to combine Post with Kao's teachings of using a DMA controller and DMA techniques to transfer boot data from nonvolatile memory to RAM; in the resulting combination, Post's control circuitry 112 would be modified to include a DMA controller (*e.g.*, Kao's DMA engine 120) that performed Post's described functions of, at least, transferring boot data pages subsequent to the first identified page from nonvolatile memory 120 to memory 114. *Id.* at 34 (citing Ex. 1003 ¶ 91). Petitioner contends that a skilled artisan would have been motivated to combine the relevant teachings of Post and Kao in this way, asserting that:

A POSITA would have understood Kao to provide a teaching, suggestion, or motivation to make this combination because Kao taught that using DMA techniques to perform memory transfers offloads work from the processor, thus reducing the processor's workload in booting an embedded system and reducing system complexity and cost.

* * *

A POSITA would have understood reducing processor workload to be a universal motivation within the field of embedded system design, and this would have further

motivated a POSITA to combine Post with Kao's teaching of using a DMA controller to perform transfers.

* * *

Kao disclosed that its DMA controller and techniques had been used to improve one device (Kao's embedded system 100, which included one or more processors and NVM 130 (INTEL-1004, 3:22-27, 3:50-51)), and a POSITA would have recognized that Kao's DMA controller and techniques would improve Post's similar device (embedded system component SoC 110, which included one or more processors and NVM (INTEL-1005, 3:33-38, 2:60-3:2)) in the same way (by offloading processor workload and reducing processor complexity and cost).

* * *

[B]oth Post and Kao addressed the problem of transferring boot data from NVM during boot of an embedded system, and Kao's DMA controller and DMA techniques were a known way to address that problem.

* * *

[A] POSITA would have viewed the combination of Post and Kao as involving the simple substitution of one known element (DMA controller and DMA techniques, as suggested by Kao) for another (using the core processor to transfer boot data from nonvolatile memory to RAM, as taught by Post) and would have understood that Post's processor-based system was ready for the improvements of reduced processor workload, system complexity, and cost.

Pet. 34–37.

Petitioner further asserts that a person of ordinary skill in the art “would also have had a reasonable expectation of success in making this combination,” and contends that “Post taught that its control circuitry 112 ‘can include any suitable components, circuitry, or logic’ and Kao demonstrated that a POSITA would have known how to implement DMA

transfers of boot data in an embedded system (Kao's) that was similar to the system of Post." *Id.* at 36 (citations omitted).

Patent Owner argues the "proposed use of Kao's DMA engine 120 is irreconcilably inconsistent with Kao's teachings" (Prelim. Resp. 12) and that "Petitioner's combination is not supported by any evidence of record (*id.* at 15).

We do not agree with Patent Owner's arguments. As an initial matter, it is not clear that Patent Owner accurately characterizes Petitioner's proposed combination, which is a combination of teachings, not the bodily incorporation of everything taught by Kao. Petitioner proposes modifying Post's control circuitry—based on the general teachings of Kao—to include a DMA controller to perform the transferring of boot data pages, which, prior to modification, would have been a function performed by Post's control circuitry. Pet. 34. Patent Owner discusses extensively how Kao purportedly uses its DMA engine, and implies that the combination must incorporate the same type of operation and further implies that this results in "irreconcilably inconsistent" aspects of Petitioner's case. *See* Prelim. Resp. 13–14; *see also, e.g., id.* at 13 (Patent Owner arguing that Kao teaches a specific boot sequence that "deviates" from the sequence in Petitioner's proposed combination). We do not understand Petitioner's proposed combination to rely on the bodily incorporation of every detail of Kao, as Patent Owner implies.

Further, even if Patent Owner is correctly characterizing Petitioner's theory, Patent Owner's arguments that Petitioner's proposed use of Kao is inconsistent with Kao's teachings and that Petitioner's modification "negates the very advantages of using the DMA engine 120 that Kao discloses" is, at

this stage, merely attorney argument on highly technical matters. Prelim. Resp. 13. Petitioner’s contentions, on the other hand, are supported, *inter alia*, by the testimony of Dr. Alpert, which, at this point, is unrebutted. *See, e.g.*, Pet. 34–38 (citing Ex. 1003 ¶¶ 91–94). As Dr. Alpert’s testimony is evidence, we disagree with Patent Owner’s argument that “Petitioner’s combination is not supported by any evidence of record.” Prelim. Resp. 15.

We determine that Petitioner has articulated sufficient reasons, for purposes of this Decision, why one of ordinary skill in the art would have modified Post’s system to include a DMA controller, and that Petitioner has demonstrated sufficiently, for purposes of this Decision, that the proposed combination yields an embedded system having all of the elements of limitation 6a.

- c. [6b] a power-on reset (POR) sequencer, wherein the POR sequencer uses a POR sequencer descriptor which is a preassembled descriptor that is stored in the nonvolatile memory,*

This limitation calls for a POR sequencer that uses a POR sequencer descriptor, which, as discussed further below in the context of limitation 6c, includes register information and DMA controller descriptors. Pertinent to this limitation 6b, Petitioner’s proposed combination relies on Wilson for register information teachings. *See, e.g.*, Pet. 41–48.

As to the recited “power-on reset (POR) sequencer,” Petitioner contends that “Post’s control circuitry 112, as modified to include Wilson’s register file and circuitry to perform the register loading functionality of Wilson’s RLA engine, facilitated a boot process for electronic device 100 after power-on and was a ‘POR sequencer.’” *Id.* at 41 (citing Ex. 1003 ¶¶ 103–106). According to Petitioner, “Post disclosed that when electronic device 100 was powered on, control circuitry 112 loaded ‘firmware, such as

a first stage bootloader’ that was used to ‘load a second stage bootloader or the kernel and operating system of electronic device 100,’” and that, “[o]nce the operating system was loaded, ‘control of circuitry 112 [w]as transferred to the operating system.’” *Id.* (citations omitted; second alteration in original); *see also id.* (asserting that “Post’s Figures 4–6 illustrated boot processes performed by control circuitry 112”).

Petitioner turns to Wilson, asserting that it discloses, *inter alia*, a microprocessor having a register file with various registers. *Id.* at 41–42. Petitioner further asserts that “Wilson taught that predefined initial values had to be stored in the registers of the register file before microcontroller 100 could assume ordinary operation,” and that “[t]hese initial register values were therefore included as ‘part of boot data that is loaded in the microcontroller during a boot-up procedure.’” *Id.* at 43 (citations omitted). Petitioner additionally asserts that “Wilson’s microcontroller included a Register Load Assist (‘RLA’) engine 104 that ‘decompress[ed] [compressed boot data] into various register initialization values’ and ‘load[ed] these values into their respective registers’ during the boot process.” *Id.* (alterations in original) (citing Ex. 1003, 6:4–7, 5:47–51, Figs. 1, 3; Ex. 1003 ¶ 112).

With regard to combining the relevant teachings of Post and Wilson, Petitioner contends that a person of ordinary skill in the art “would have found it obvious to modify Post in view of Wilson’s teachings, such that Post’s control circuitry 112 included Wilson’s register file and circuitry to perform the Register Load Assist functionality of the RLA engine.” *Id.* at 43. The Petition states:

As Post is silent about techniques for loading initialization values into its registers during boot, a POSITA would have

looked to Wilson for implementation details on how to do so. (INTEL-1003, ¶ 114.)

* * *

A POSITA would have been motivated to similarly custom design Post's SoC 110, including control circuitry 112, to realize the same benefits of performing specific predefined functions for configuring a mobile phone, which required initialization of configuration registers as taught by Wilson. (INTEL-1003, ¶ 115.) Post taught that its control circuitry "can include any suitable components, circuitry, or logic" (INTEL-1005, 3:36–38), which a POSITA would have viewed as enabling and encouraging the incorporation of circuitry such as that described by Wilson. (INTEL-1003, ¶ 115.)

Further, a POSITA would have viewed Wilson as providing a teaching, suggestion, or motivation to include registers and Register Load Assist functionality in Post's control circuitry 112 because Wilson taught that register values must be loaded before a microcontroller can begin normal operation. (INTEL-1013, 4:59–61 ("for microcontroller 100 to commence ordinary operation, it can be required that certain predefined values be loaded in the registers of register file 102"); INTEL-1003, ¶ 116.) Wilson also taught that including initial register information along with other boot data, such as compressed initial memory values and instructions, could provide the benefit of improving the speed of a boot process, which a POSITA would have viewed as a further teaching, suggestion, or motivation for this combination. (*Id.*, 5:10–11, 5:25–26, 5:46–47, 6:42–44; INTEL-1003, ¶ 117.)

* * *

[A] POSITA would have been motivated to make this combination because Wilson's teaching of including compressed initial register values in boot data, and using the Register Load Assist circuitry to decompress those values and load the microcontroller's registers with those values, was a known technique for solving the known problem of how to initialize processor register information at bootup (INTEL-1013, 1:25–28, 2:54–63, 6:27–41, Figs. 1, 3; INTEL-1016, 555); and including register initialization information in Post's boot data would have been within the skill of an ordinary

artisan, because they would be generally familiar with processor registers and processor boot techniques (INTEL-1009, 137; INTEL-1013, 1:25–28; INTEL-1005, 1:14–26), and because Wilson showed that a POSITA knew how to do so for a microcontroller similar to Post’s SoC. (INTEL-1003, ¶ 119, 121.)

* * *

Further, a POSITA would have recognized the systems of Post and Wilson to have various similarities, which would have prompted them to look to Wilson as related teachings. . . . Both Post and Wilson taught storing boot data of an embedded system in an external nonvolatile memory. (INTEL 1005, 1:32–34; INTEL-1013, 1:57–60; INTEL-1003, ¶ 120.) In addition, Wilson, like Post, was directed to improving the boot process in an embedded system. (INTEL-1013, 2:4–8; INTEL-1005, 1:35–38; INTEL-1003, ¶ 120.) Further, Wilson, like Post, taught booting an embedded system within a mobile phone featuring a multifunction touch screen. (INTEL-1013, 3:35–37; INTEL-1005, 2:60–67; INTEL-1003, ¶ 120.)

Pet. 43–47.

Petitioner further asserts that a person of ordinary skill in the art “would also have had a reasonable expectation of success in making this combination, because POSITAs were familiar with processor registers and techniques for initializing registers at system bootup.” *Id.* at 45 (citing Ex. 1016, 137, 156, 555; Ex. 1013, 1:24–28; Ex. 1003 ¶ 118).

The remaining portion of limitation 6b recites, “the POR sequencer uses a POR sequencer descriptor which is a preassembled descriptor that is stored in the nonvolatile memory.” Petitioner contends that “[t]he boot data (including initial register values) and associated metadata of the asserted combinations facilitated a boot process and were thus a *POR sequencer descriptor*.” Pet. 56. Petitioner further contends that Post teaches the recited “preassembled descriptor that is stored in the nonvolatile memory,” asserting

that “Post taught storing boot data and metadata in a linked-list format in nonvolatile memory 120” and that the linked-list format includes next page and prior page pointers. *Id.* at 59 (citing Ex. 1005, 1:57–64, 4:46–47, 5:39–40, 6:19–24, 6:28–31, 6:38–42; Ex. 1003 ¶¶ 143–144). Petitioner additionally contends that, in the proposed combinations, the POR sequencer uses this data. *Id.* at 57–59. Specifically, Petitioner contends that, “[i]n the asserted combinations, the *POR sequencer used* boot data (part of ‘*the POR sequencer descriptor*’) to load a second stage bootloader and device kernel and operating system and load values into registers,” and “also *used* metadata pointers (also part of ‘*the POR sequencer descriptor*’) to fetch each fragment of boot data stored in NVM 120 and copy it to memory 114.” *Id.* at 57–58 (citations omitted).

Patent Owner argues that “the resulting combination fails to disclose a POR sequencer performing the claimed functions as recited in claim 1.” Prelim. Resp. 15. Specifically, Patent Owner argues that “Petitioner’s adoption of Kao’s DMA engine 120 for the purported benefits of that DMA engine 120 as discussed in the Petition (*see* Pet., 34–38) confirms that Post’s control circuitry 112 would not ‘detect, in the nonvolatile memory, a descriptor used in a boot process and configured to copy the descriptor from the nonvolatile memory to the RAM,’ as claimed.” *Id.* at 14. We do not agree with this argument at least because the quoted “detect . . . and configured to copy” language from limitation 1b does not appear in independent claim 6. The corresponding limitation 6b merely recites that the “the POR sequencer uses a POR sequencer descriptor.” Ex. 1001, 10:4–5.

Also, to the extent that Patent Owner argues that the subject functions performed by Post’s control circuitry are not utilized in Petitioner’s proposed combination, we disagree. *See* Prelim. Resp. 12–13 (Patent Owner asserting that Petitioner’s contention that Post discloses a POR sequencer “ignores Petitioner’s use of Kao’s DMA engine 120 in lieu of Post’s control circuitry 112 for the purpose of transferring boot data pages from nonvolatile memory 120 to memory 114 in the relied-upon combination of references.”). We understand Petitioner to contend that, in the proposed combination, Post’s control circuitry is modified, not replaced entirely, and that the performance of certain functions is shifted over from Post’s processor to a DMA controller. *See* Pet. 34 (“[I]n the resulting combination, Post’s control circuitry 112 would be modified to include a DMA controller (*e.g.*, Kao’s DMA engine 120) that performed Post’s described functions of, at least, transferring boot data pages subsequent to the first identified page from nonvolatile memory 120 to memory 114.”).

We determine that Petitioner has demonstrated sufficiently, for purposes of this Decision, that the proposed combination contains “a power-on reset (POR) sequencer, wherein the POR sequencer uses a POR sequencer descriptor which is a preassembled descriptor that is stored in the nonvolatile memory,” as recited in limitation 6b.

d. [6c] wherein the POR sequencer descriptor includes register information and DMA controller descriptors, and

As discussed for the previous limitation, Petitioner contends that boot data is part of the recited POR sequencer descriptor. Pet. 56. Petitioner contends that “Wilson disclosed initial register values ([the recited] ‘register information’) included in the boot data.” *Id.* at 60.

Petitioner also contends that a person of ordinary skill in the art “would further have understood, and found it obvious, that Post’s next-page pointers (also part of the *POR sequencer descriptor*) would be *DMA controller descriptors* in the asserted combinations.” *Id.* at 61 (citing Ex. 1003 ¶¶ 146–150). Petitioner asserts that:

As discussed for claim 1[a], a POSITA would have found it obvious for Post’s control circuitry 112 to include *a DMA controller* (e.g., Kao’s DMA engine 120) that performed Post’s described functions of, at least, transferring pages of boot data subsequent to the first page from nonvolatile memory 120 to memory 114. (*Supra* Sec. VI.B.2.) The DMA controller of the asserted combinations would have used Post’s next-page pointer metadata as the “descriptors” to enable these transfer operations. (INTEL-1003 ¶ 149.)

Id. Petitioner discusses Post’s pointer teachings, compares Post’s Figure 3 to Figure 5 of the ’694 patent, and contends that “[t]he ’694 patent describes DMA controller descriptors in this same way, as pointers pointing to additional locations in nonvolatile memory.” *Id.* at 62–63 (citing, *inter alia*, Ex. 1003 ¶ 147).

Petitioner further contends that a person of ordinary skill in the art “would have found it obvious for Kao’s DMA controller to use Post’s next-page pointer metadata as a ‘DMA controller descriptor,’ and that “Kao supplied a teaching, suggestion, or motivation for a DMA controller to use the address of an item in NVM—e.g., the information in Post’s next-page pointer metadata—to facilitate a transfer.” Pet. 64 (citations omitted).

Petitioner additionally contends that:

A POSITA would also have understood this to be the combination of known elements (Kao’s DMA controller and Post’s next-page pointer information) according to known methods (Kao’s DMA techniques of identifying data or instructions located at a particular address in NVM), to yield

predictable results (identification of pages of boot data in Post's NVM 120 and transfer of the boot data to RAM 114).

Id. at 65 (citing Ex. 1003 ¶ 150).

Patent Owner argues that “Petitioner incorrectly interprets the claimed ‘DMA controller descriptors’ as Post’s pointers, and overlooks that Kao’s DMA engine does not use or need any DMA controller descriptors to be included as part of a POR sequencer descriptor.” Prelim. Resp. 18. We do not agree with Patent Owner’s argument. Patent Owner attacks Petitioner’s showing by arguing the references individually, rather than addressing the combination of teachings relied upon in the Petition. Specifically, Patent Owner argues that Petitioner has not shown that Post discloses the use of DMA components, for which Petitioner relies on Kao, and Patent Owner argues that Kao does not disclose a descriptor, for which Petitioner relies on Post. *See id.* at 18 (arguing that Petitioner “incorrectly (and illogically) alleges that Post discloses a DMA controller descriptor”), 20 (“Petitioner does not and cannot identify any DMA controller descriptor in Kao.”).

Regarding Petitioner’s contention that Post’s next-page pointers would be DMA controller descriptors in the asserted combinations, Patent Owner acknowledges that “the ’694 Patent discloses that a last DMA controller descriptor []of a POR sequencer descriptor ‘points to the next POR sequencer descriptor fragment 502,’” but argues that “this is not the entirety of a DMA controller descriptor as disclosed in the ’694 Patent.” *Id.* at 20. However, Patent Owner does not clearly state its contention as to the proper construction of “DMA controller descriptor” and does not clearly identify what, beyond a pointer, purportedly is required by that descriptor. *See* Prelim. Resp. 10 (Patent Owner arguing that “Petitioner’s proposed

characterization of the claimed ‘DMA controller descriptors’ as satisfied by Post’s pointer is incorrect, and does not comport with the plain and ordinary meaning of this claim term,” but not stating the purported plain and ordinary meaning); *see also id.* at 18–20.

We determine that Petitioner has demonstrated sufficiently, for purposes of this Decision, that the proposed combination contains a “POR sequencer descriptor [that] includes register information and DMA controller descriptors,” as recited in limitation 6c.

e. [6d] wherein the POR sequencer reads and uses the register information to update configuration registers of the embedded system.

Petitioner contends that “[t]he control circuitry of the asserted combinations *read and used the register information to update configuration registers of the embedded system.*” Pet. 65. According to Petitioner, “Wilson taught loading initial register values into a microcontroller’s registers,” and “Wilson’s registers included registers that were ‘custom designed to perform specific predefined functions,’ such as configuring the touch screen display and pixel color values (‘configuration registers’).” *Id.* at 65–66 (citations omitted). Petitioner further contends that “[t]he ’694 patent similarly describes a ‘register initialization phase’ that entails loading registers with initial values.” *Id.* at 66 (citing Ex. 1001, 7:17–44, Fig. 7; Ex. 1003 ¶ 156). Petitioner additionally contends that a person of ordinary skill in the art “would have found it obvious for control circuitry 112, as modified to include Wilson’s register file and circuitry to perform functionality of Wilson’s RLA engine (‘POR sequencer’) to read and use register information to update configuration registers of Post’s SoC 110 (‘embedded system’) as part of the boot-up process.” *Id.* at 66 (citing

Ex. 1003 ¶¶ 151–157; referring to the reasons to combine described in Section VI.B.3 of the Petition).

Patent Owner does not raise any arguments regarding Petitioner’s contentions beyond those addressed above. We determine that Petitioner has demonstrated sufficiently, for purposes of this Decision, that, in the proposed combination, “the POR sequencer reads and uses the register information to update configuration registers of the embedded system,” as recited in limitation 6d.

f. Conclusion as to Independent Claim 6

Having considered the parties’ arguments and evidence, we determine that Petitioner has demonstrated a reasonable likelihood of prevailing in its challenge to claim 6 as being obvious over Post, Wilson, and Kao and over Post, Wilson, Kao, and So.

III. CONCLUSION

Petitioner has demonstrated a reasonable likelihood of prevailing in showing the unpatentability of independent claim 6. Accordingly, we institute an *inter partes* review on all challenged claims on all asserted grounds. *See SAS Inst. Inc. v. Iancu*, 138 S. Ct. 1348, 1354, 1359–60 (2018); 37 C.F.R. § 42.108(a) (“When instituting *inter partes* review, the Board will authorize the review to proceed on all of the challenged claims and on all grounds of unpatentability asserted for each claim.”). At this stage of the proceeding, we have not made a final determination with respect to the construction of any claim term or with respect to the patentability of any of the challenged claims.

IV. ORDER

For the foregoing reasons, it is

ORDERED that, pursuant to 35 U.S.C. § 314(a), an *inter partes* review of claims 1, 5, 6, 9, and 12 of the '694 patent is instituted with respect to all grounds of unpatentability set forth in the Petition; and

FURTHER ORDERED that, pursuant to 35 U.S.C. § 314(c) and 37 C.F.R. § 42.4(b), *inter partes* review of the '694 patent shall commence on the entry date of this Order, and notice is hereby given of the institution of a trial.

IPR2023-00787
Patent 10,120,694 B2

FOR PETITIONER:

Christina McCullough
Lori A. Gordon
Christopher Kelley
Perkins Coie LLP
Intel-BitMicro-084IPRService@perkinscoie.com
mccullough-ptab@perkinscoie.com
gordon-ptab@perkinscoie.com
kelley-ptab@perkinscoie.com

FOR PATENT OWNER:

Wayne M. Helge
James T. Wilson
Nossaman LLP
whelge@nossaman.com
Salvarado@nossaman.com