

UNITED STATES PATENT AND TRADEMARK OFFICE

---

BEFORE THE PATENT TRIAL AND APPEAL BOARD

---

INTEL CORPORATION,  
Petitioner,

v.

BiTMICRO, LLC,  
Patent Owner.

---

Case IPR2023-00785  
Patent 9,858,084

---

**PATENT OWNER'S PRELIMINARY RESPONSE**

**TABLE OF CONTENTS**

|      |                                                                                                                                                                                                                                                     |    |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| I.   | INTRODUCTION .....                                                                                                                                                                                                                                  | 1  |
| A.   | Prior to Activating a CPU, the Disclosed Invention Loads a<br>POR Sequencer Descriptor Comprising Information to Initialize<br>Configuration Registers, Direct Memory Access (DMA)<br>Descriptors, and System Firmware for the Boot Process.....    | 2  |
| B.   | The Challenged Claims and Petitioner’s Challenges .....                                                                                                                                                                                             | 7  |
| II.  | PETITIONER’S CLAIM CONSTRUCTION IS PLAINLY INCORRECT<br>AND FATAL TO THE PETITION .....                                                                                                                                                             | 11 |
| III. | PETITIONER’S CHALLENGES FAIL ON THE MERITS .....                                                                                                                                                                                                    | 13 |
| A.   | Ground 1 based on Post with Kao and Wilson Fails.....                                                                                                                                                                                               | 13 |
| i.   | Petitioner’s Proposed Combination of Post with Wilson is<br>Inconsistent with Petitioner’s Asserted Reason to Form the<br>Combination, and Fails to Disclose Every Claimed Feature .....                                                            | 13 |
| ii.  | Petitioner’s Proposed Combination Fails to Disclose or<br>Suggest a POR Sequencer Descriptor that Includes Direct<br>Memory Access (DMA) Descriptors, as Claimed .....                                                                              | 16 |
| iii. | Petitioner’s Proposed Combination of Post, Kao, and Wilson<br>Also Fails to Disclose or Suggest the Steps of Verifying an<br>Integrity of the POR Sequencer Descriptor and Detecting<br>Any Error in the POR Sequencer Descriptor, as Claimed ..... | 22 |
| B.   | Ground 1’s Challenge Fails to Establish the Unpatentability of<br>Claim 1 .....                                                                                                                                                                     | 25 |
| IV.  | CONCLUSION.....                                                                                                                                                                                                                                     | 26 |

Pursuant to 37 C.F.R. § 42.107, Patent Owner BiTMICRO, LLC (“BiTMICRO” or “Patent Owner”) files this preliminary response to the Petition, setting forth reasons why the Petition for *inter partes* review (“IPR”) of claims 1, 5, 8, 9, 19, and 20 of U.S. Patent No. 9,858,084 (the “’084 Patent”), as requested by Intel Corporation (“Petitioner”), should be denied.<sup>1</sup>

## **I. INTRODUCTION**

The ’084 Patent, entitled “Copying of Power-On Reset Sequencer Descriptor from Nonvolatile Memory to Random Access Memory,” relates to a method and apparatus for booting an embedded system from a storage device, “prior to activating a CPU.” ’084 Pat., Abstr. In particular, the ’084 Patent describes an exemplary power-on reset (POR) sequencer that “controls the overall boot process of the embedded system” from nonvolatile memory and loads a POR sequence descriptor “to update the configuration registers of the system and to enable CPU-independent data transfers with the use of DMA controllers.” ’084 Pat., Abstr, 2:52-58. The exemplary POR sequencer’s loading of DMA controller descriptors enables the initialization of associated DMA controllers, which can then take control of the CPU bus to perform memory transactions independently of the

---

<sup>1</sup> By submitting this Preliminary Response, no waiver of any argument is intended by Patent Owner. Patent Owner will have a right to file a complete response to the Petition, should the Board institute *inter partes* review. 37 C.F.R. § 42.120(a).

embedded system's CPU(s) and therefore conserve processing resources for other activities.

**A. Prior to Activating a CPU, the Disclosed Invention Loads a POR Sequencer Descriptor Comprising Information to Initialize Configuration Registers, Direct Memory Access (DMA) Descriptors, and System Firmware for the Boot Process**

As illustrated in FIG. 2 of the '084 Patent, an exemplary embedded system includes one or more central processing units (CPUs), a reset controller, a storage device controller, one or more direct memory access (DMA) controllers, a random access memory (RAM), a nonvolatile memory controller, a debug interface, and a Power-On Reset (POR) Sequencer. *See, e.g., '084 Pat., Fig. 2.*<sup>2</sup>

---

<sup>2</sup> For the Board's reference, the FIG. 2 included in Ex. 1001 is not the complete drawing, as certain components including CPUs 201, the optional DMA controllers 203, and reset controller 206 in Ex. 1001's FIG. 2 appear to be annotated out of the drawing. The Petition includes a version of FIG. 2 that does not omit these components. *See Pet., 13.*

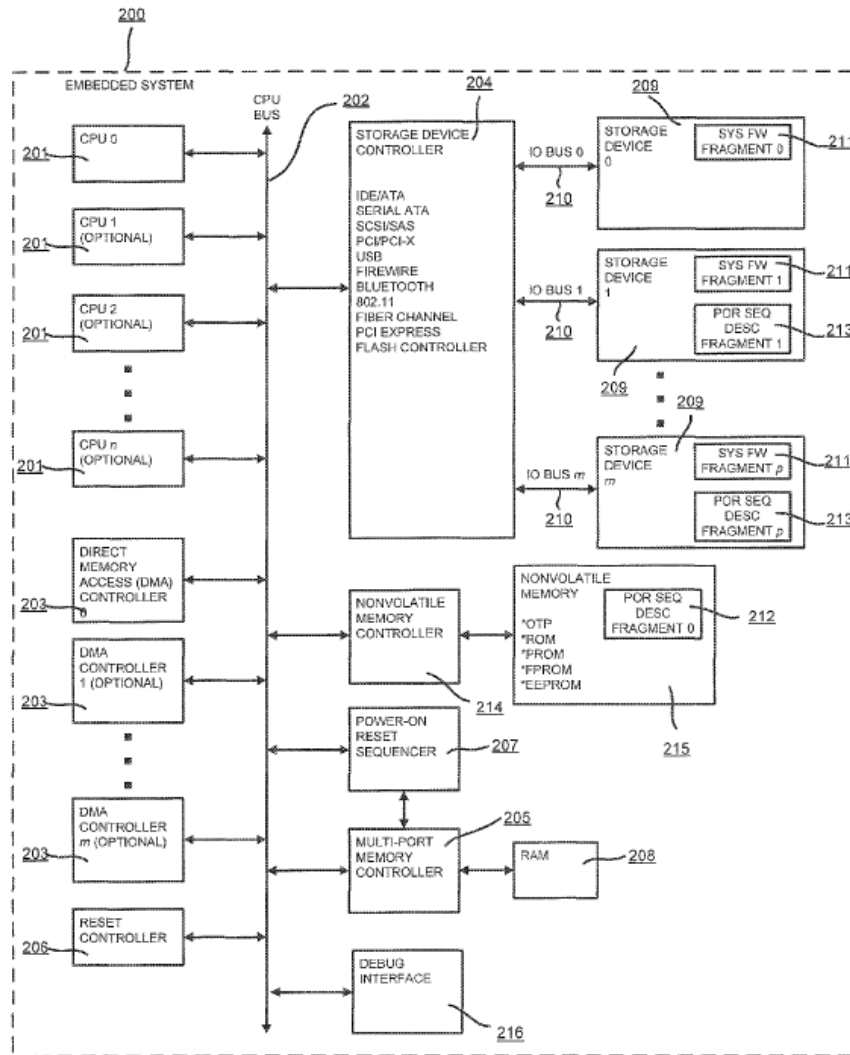


FIG. 2

Although conventional embedded systems could use a CPU to initiate and/or control the boot process, the '084 Patent discloses a method of detecting and loading a POR sequencer descriptor prior to activating a CPU. In this exemplary method, the POR sequencer detects, in the nonvolatile memory, a descriptor used in a boot process and copies the POR sequencer descriptor from the nonvolatile memory to the RAM for use in the boot process.

As illustrated in Figure 4 of the '084 Patent, reproduced below, an exemplary POR sequencer descriptor includes register information for a controller core, control information and DMA controller descriptors to activate the DMA controllers, and a POR sequencer descriptor checksum to not only ensure that the entirety of the POR sequencer descriptor is loaded into RAM, but to verify the integrity of the data and code once loaded into RAM. *See, e.g.*, '084 Pat., 7:2-5 (“POR sequencer 207 then checks the integrity of POR sequencer descriptor 212 [loaded into RAM 208] by comparing the computed checksum to the POR sequencer descriptor checksum 507.”).

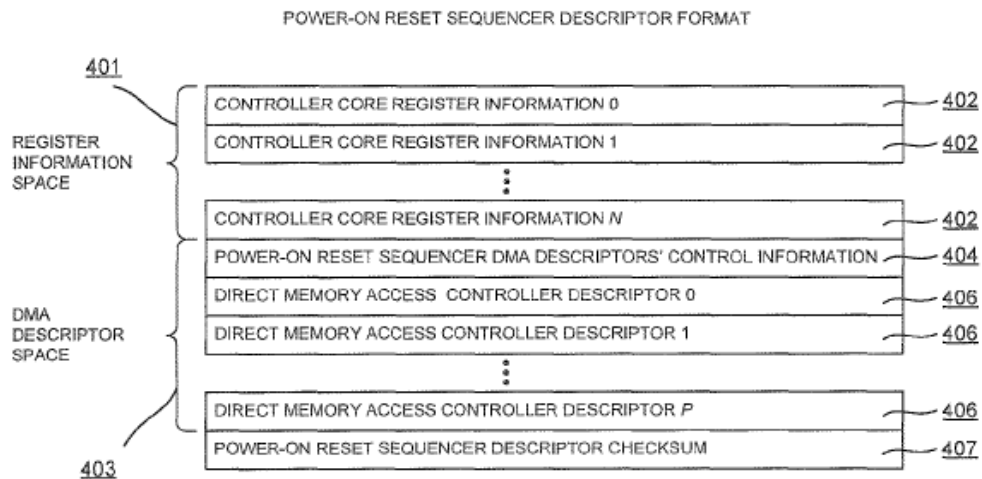
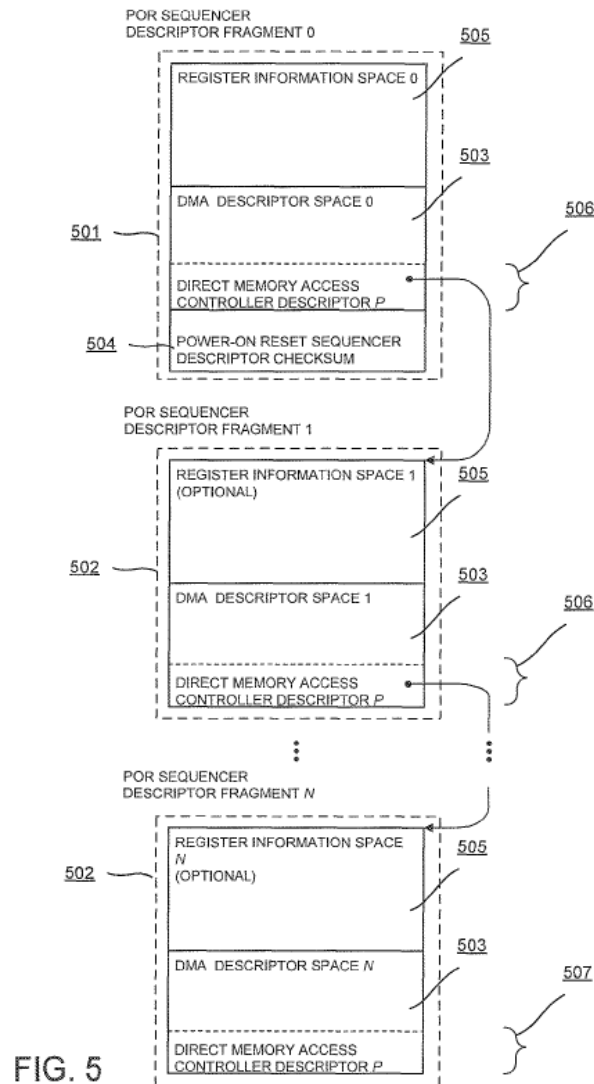


FIG. 4

The '084 Patent also discloses that the entire POR sequencer descriptor may not be stored in a single block or location in the nonvolatile memory. In this case,

the '084 Patent uses POR sequencer descriptor fragments, where “the descriptor is divided into fragments and distributed across the nonvolatile memory and storage devices attached to the system.” '084 Pat., 6:34-37. In this case, the '084 Patent discloses the option to store register information space fragments, DMA descriptor space fragments, and DMA controller descriptor fragments in each POR sequencer descriptor fragment, with the POR sequencer descriptor checksum stored in the first POR sequencer descriptor fragment. In order for the POR sequencer to locate the next POR sequencer descriptor fragment, the '084 Patent discloses that the last DMA controller descriptor in a POR sequencer descriptor fragment also “points to the next POR sequencer descriptor fragment.” '084 Pat., 6:44-46. This arrangement of POR sequencer descriptor fragments, and use of the DMA descriptor space fragments to point to the next POR sequencer descriptor fragment, is illustrated in FIG. 5 reproduced below.



This exemplary use of POR sequencer descriptor fragments is relevant to the claims challenged by Petitioner here. As claimed in the '084 Patent, the POR sequencer descriptor copied from nonvolatile memory to a random access memory (RAM) “comprises information to initialize configuration registers of the embedded system, Direct Memory Access (DMA) descriptors used to fetch other POR sequencer descriptor fragments, and a system firmware.” '084 Pat., 9:18-22.

Additionally, the computer-implemented method includes “verifying an integrity of the POR sequencer descriptor,” and “detecting any error in the POR sequencer descriptor.” ’084 Pat., 9:25-27. This method “allows the embedded system to verify an integrity of the user data and program code used in a boot process in order to provide a resilient boot up sequence.” ’084 Pat., 9:28-31.

**B. The Challenged Claims and Petitioner's Challenges**

Consistent with these disclosures, the challenged claims are directed generally to a method of performing a boot process of an embedded system using a power-on reset (POR) sequencer descriptor, and a related article of manufacture.

Claim 1 is the focus of Petitioner's challenges, and is reproduced below:

1. A computer-implemented method, comprising:  
releasing components of an embedded system from reset;  
detecting a power-on reset (POR) sequencer descriptor in a  
nonvolatile memory; wherein the POR sequencer descriptor  
comprises information to initialize configuration registers of the  
embedded system, Direct Memory Access (DMA) descriptors  
used to fetch other POR sequencer descriptor fragments, and a  
system firmware;  
copying the POR sequencer descriptor from the nonvolatile  
memory to a random access memory (RAM);  
verifying an integrity of the POR sequence descriptor;  
detecting any error in the POR sequence descriptor;  
wherein the computer-implemented method allows the embedded  
system to verify an integrity of user data and program code

used in a boot process in order to provide a resilient boot up sequence and wherein the computer-implemented method reduces an amount of the nonvolatile memory used in the boot process.

’084 Pat., 9:14-33. Petitioner asserts a single obviousness-based ground against claim 1 and the additional challenged claims, claims 5, 8, 9, 19, and 20. *See* Pet., 6 (identifying one Ground for Challenge). Of these additional challenged claims, claims 5, 8, and 9 depend from claim 1, claim 19 is independent (“An article of manufacture ...”), and claim 20 depends from claim 19. For the purpose of Petitioner’s challenges, claim 19 is substantially similar to claim 1 as discussed in Section III below.

In the Petition, Petitioner contends that all challenged claims are obvious over a combination of the Post<sup>3</sup> reference, the Kao<sup>4</sup> reference, and the Wilson<sup>5</sup> reference. However, Petitioner’s challenges fail for numerous reasons.

First, Petitioner’s alleged reason for combining Post and Wilson is inconsistent with the resulting combination of these references. Petitioner proposes to incorporate Wilson’s register information into Post’s boot code so that Wilson’s register information can be loaded by Post’s control circuitry 112 upon startup.

---

<sup>3</sup> U.S. Patent No. 8,799,555 to Post *et al.* (“Post”) (Ex. 1005).

<sup>4</sup> U.S. Patent No. 7,490,177 to Kao (“Kao”) (Ex. 1004).

<sup>5</sup> U.S. Patent No. 7,861,073 to Wilson (“Wilson”) (Ex. 1013).

But Petitioner also contends that Wilson's register information *must* be pre-loaded before Post's control circuitry 112 can perform normal operations. These positions cannot be reconciled. Post's control circuitry 112 cannot be asked to load the very register information that is necessary for that same control circuitry 112 to perform its normal operations. If Wilson's registration information must first be loaded for Post's control circuitry 112 to operate, then Petitioner cannot rely on Post's control circuitry 112 to perform that prerequisite loading of its own register information. As a result, in Petitioner's proposed combination of Post and Wilson, there is no mechanism to pre-load register information that is necessary for Post's control circuitry 112 to perform normal operations. This combination as advanced in the Petition is illogical and cannot stand up to scrutiny.

Second, Post fails to disclose any DMA controller, and thus Petitioner incorrectly asserts that Post discloses the claimed "Direct Memory Access (DMA) descriptors" as part of the claimed "POR sequencer descriptor." Not only is this assertion wrong, but Petitioner makes the error of correlating the claimed "DMA descriptor" to nothing more than a pointer in Post. While the '084 Patent discloses that a DMA controller descriptor fragment can *include* a pointer to point to a next POR sequencer descriptor fragment, as introduced above, the '084 Patent also discloses that the DMA descriptor is written by the POR sequencer to a "configuration space of an available DMA controller" to activate that DMA

controller. '084 Pat., 7:52-61. Petitioner ignores this disclosure in its incorrect argument that the claimed "DMA descriptor" is disclosed by a pointer in Post (which again, does not disclose a DMA controller, much less a DMA descriptor). Furthermore, Petitioner overlooks that Kao does not disclose a DMA descriptor either, and does not disclose copying a DMA descriptor to RAM in a boot-up process. Rather, Kao pre-stores control code in a NAND flash memory before booting to avoid having to load code for activating a DMA controller as part of the boot-up process. As a result, Kao has no DMA descriptors to be incorporated into Post, as alleged by Petitioner. For all these reasons, Petitioner's proposed combination fails to disclose or suggest any DMA descriptors as part of the POR sequencer descriptor claimed in claim 1.

Third, Petitioner once again incorrectly relies on Post's use of pointers in code fragments, this time to disclose the claimed requirement of the POR sequencer "verifying an integrity of the POR sequencer descriptor." While Post discloses the use of pointers to ensure that every page of boot code is located or mapped, Post does not disclose any verification of the integrity of a POR sequencer descriptor. Further, while Post discloses using error correction codes with Post's individual boot code pages, this is to determine whether the page includes boot code by checking for a codeword or a program status. Post is clear that this error correction is to ensure that an identified page still in non-volatile

memory contains boot data. These steps of using pointers and ensuring that a located boot code page possesses boot code does not verify the integrity of a POR sequencer descriptor, and does not detect for errors within the POR sequencer descriptor, as claimed.

For these reasons, as more fully explained below, the sole Ground presented in the Petition is deficient. The Petition should be denied without institution.

## **II. PETITIONER’S CLAIM CONSTRUCTION IS PLAINLY INCORRECT AND FATAL TO THE PETITION**

In the Petition, Petitioner contends that no terms require an express construction, and purports to be applying the plain and ordinary meaning of the claim terms. *See* Pet., 17. However, Petitioner’s proposed characterization of the claimed “Direct Memory Access (DMA) descriptors” as satisfied by nothing more than a pointer in Post (which does not disclose a DMA controller or a DMA descriptor) is incorrect and does not comport with the plain and ordinary meaning of this claim term. Further, this incorrect characterization fails to consider the ’084 Patent’s specification disclosure that the “Direct Memory Access (DMA) descriptors” allow for activation of associated DMA controllers, and is inconsistent with Petitioner’s own evidence.

The claim language at issue recites, in relevant part, a “POR sequencer descriptor comprises ... Direct Memory Access (DMA) descriptors used to fetch other POR sequencer descriptor fragments ... .” ’084 Pat., 9:18-22. Petitioner’s

approach incorrectly focuses only on the latter portion of this feature—“used to fetch other POR sequencer descriptor fragments”—and ignores that the claim term also requires a DMA descriptor, which is more than Post’s pointer.

Without waiving its right to submit additional proposed constructions in a Patent Owner Response, should one be necessary, Patent Owner submits that the Board need not engage in a formal construction of “Direct Memory Access (DMA) descriptors” in this pre-institution phase. All evidence of record confirms that a descriptor is more than just Post’s pointer, and Petitioner’s decision to disregard the plain meaning of this claim term is fatal to the Petition. Additional discussion of the proper interpretation of the claimed “DMA descriptors,” and why Petitioner is wrong to characterize this feature as disclosed by Post’s pointer, is provided in the context of Patent Owner’s substantive arguments on this term in Section III.A.ii below.

For this reason and the additional reasons presented in Section III below, Petitioner’s Ground includes fundamental flaws, and fails to establish the obviousness of any challenged claim.

### **III. PETITIONER'S CHALLENGES FAIL ON THE MERITS**

#### **A. Ground 1 based on Post with Kao and Wilson Fails**

##### **i. Petitioner's Proposed Combination of Post with Wilson is Inconsistent with Petitioner's Asserted Reason to Form the Combination, and Fails to Disclose Every Claimed Feature**

As noted above, claim 1 recites, in relevant part, “detecting a power-on reset (POR) sequencer descriptor in a nonvolatile memory; wherein the POR sequencer descriptor comprises information to initialize configuration registers of the embedded system ... ;” and “copying the POR sequencer descriptor from the nonvolatile memory to a random access memory (RAM).” Petitioner's theory for how the asserted combination of Post and Wilson allegedly discloses these claimed features is inconsistent with Petitioner's very basis to form the combination, and fails.

In challenging claim 1, Petitioner notes that Post's control circuitry 112 is first “booted by executing program code from boot ROM.” Pet., 28 (citing Post, 3:41-43) (“[A]t bootup of electronic device 100, control circuitry 112 may operate under the control of program code stored in boot ROM 116.”). Petitioner then discusses that this “program code may be executed [by control circuitry 112] to begin loading additional information from NVM 120 into memory 114.” Pet., 29, 30-31; Post, 3:55-65. Notably, under Petitioner's theory and asserted combination, it is Post's control circuitry 112 that loads Post's boot code from NVM 120 into memory 114.

Addressing the relevant claim 1 language quoted above, Petitioner concedes that Post fails to disclose or suggest a POR sequencer descriptor that comprises “information to initialize configuration registers of the embedded system.” Pet., 34 (“Post is silent about techniques for loading initialization values into its registers during boot.”). Petitioner therefore relies upon Wilson to disclose the claimed “information to initialize configuration registers of the embedded system” as part of a “POR sequencer descriptor.” Petitioner alleges that it would have been obvious to include Wilson’s “information to initialize configuration registers of control circuitry 112 (including its processors)” in “Post’s boot data in NVM 120.” Pet., 34.

This argument is wholly illogical, however, because Petitioner also relies on Wilson to teach that “many microcontrollers required a boot-up process that allowed the microcontroller to ‘perform various actions to prepare [it] for ordinary operation, such as load initialization values in its registers.’” Pet., 33 (quoting Wilson, 1;24-28). Wilson explains in more detail: “The microcontroller can *require* that certain specific predefined initial values are stored in one or more of the registers of the register file *before* the microcontroller assumes ordinary operation .... [T]hese initial values can be a part of boot data that is loaded in the microcontroller during a boot-up procedure.” Wilson, 2:53-63; Pet. 33 (quoting the same) (emphasis added). In its argument in support of its attempt to combine

Post and Wilson, Petitioner even relies on Wilson to disclose that “register values *must* be loaded before a microcontroller can begin its normal operation.” Pet., 35 (emphasis in original).

Summarizing Petitioner’s theory, in order for Post’s control circuitry 112 (including processors) to begin normal operations—such as detecting and copying a POR sequencer descriptor from nonvolatile memory to RAM—control circuitry 112’s registers values must first be loaded and initialized for Post’s processors to operate normally. But Petitioner asks Post’s control circuitry 112 to perform the operations of loading boot code from NVM 120 to memory 114 before its own registers are loaded or initialized. Under these circumstances, according to Petitioner, Post’s control circuitry 112 cannot begin normal operations to pre-load its own initialized register information. Indeed, Petitioner offers no explanation how Post’s control circuitry 112 can begin the operation of loading boot code into memory 114 without the requisite initialized register values for control circuitry 114 first being loaded. This bootstrapping theory for how control circuitry 112 can prepare itself for its own normal operation is the result of a hindsight-based recreation of the challenged claim language, and fails as illogical.

Petitioner’s proposed combination improperly mixes pre- and post-boot operations, and is not supported by the evidence of record. As such, this asserted combination fails to disclose at least “detecting a power-on reset (POR) sequencer

descriptor in a nonvolatile memory; wherein the POR sequencer descriptor comprises information to initialize configuration registers of the embedded system ...” as recited in claim 1.

**ii. Petitioner’s Proposed Combination Fails to Disclose or Suggest a POR Sequencer Descriptor that Includes Direct Memory Access (DMA) Descriptors, as Claimed**

Claim 1 also recites, in relevant part, “wherein the POR sequencer descriptor comprises ... Direct Memory Access (DMA) descriptors used to fetch other POR sequencer descriptor fragments... .” Petitioner’s challenge of this claim element also fails because Petitioner improperly characterizes the claimed “descriptors” as a pointer in Post (which does not disclose a DMA controller, much less DMA descriptors), and because Kao’s DMA engine also does not disclose or use DMA descriptors. The Petition fails to establish that these claimed features are present or suggested by Petitioner’s proposed combination of Post, Kao, and Wilson.

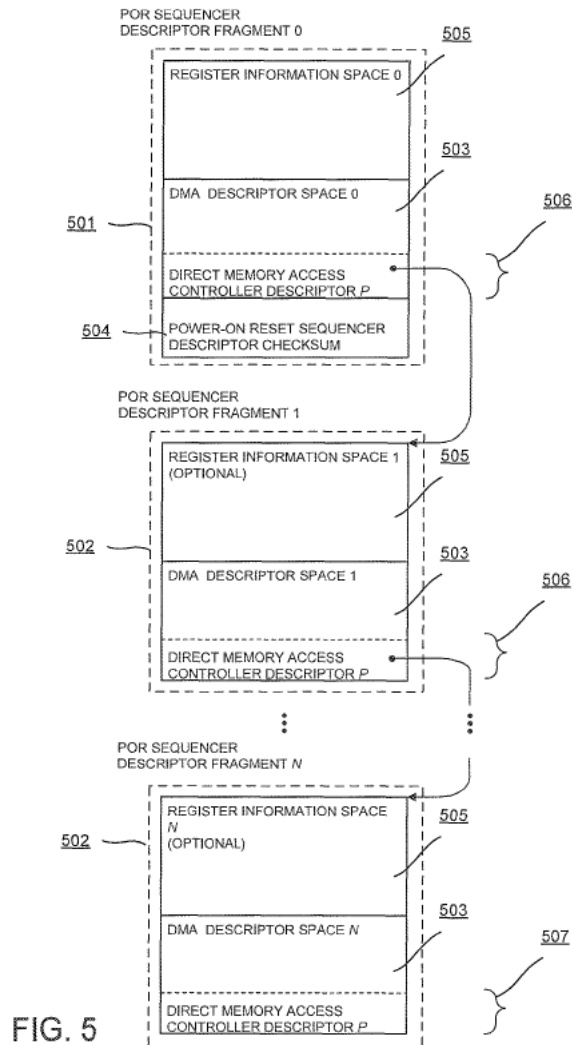
In its challenge of this claimed element, Petitioner introduces Post’s use of boot code pages and pointers to refer to a next page in a sequence. *See* Pet., 37-41. Without support, and contrary to the record evidence, Petitioner then concludes that Post’s “metadata pointers” are “descriptors.” Pet., 41 (“In sum, Post discloses metadata pointers (‘descriptors’) stored in nonvolatile memory that were used to fetch additional fragments of boot data ... .”) (emphasis omitted). Post’s pointers are not DMA descriptors, and are not included in any DMA descriptors in any

asserted reference or combination of references. Petitioner's assertion to the contrary is contradicted by the record evidence.

As disclosed in the '084 Patent, the "DMA descriptors" included in the "POR Sequencer descriptor" "are loaded by the POR Sequencer to the DMA controllers of the system." '084 Pat., 2:52-54, 2:56-58. Similarly, the "DMA controller descriptors 406" are identified by DMA descriptors' control information 404, and "are loaded by the POR Sequencer to the DMA controller's configuration register space." '084 Pat., 6:28-30.<sup>6</sup> In Figure 5, reproduced below, the DMA controller descriptors include a plurality of DMA controller descriptors, DMA controller descriptor 0 to DMA controller descriptor P. See '084 Pat., 6:40-42.

---

<sup>6</sup> As confirmed by these substantially similar disclosures, the '084 Patent uses the terms "DMA descriptors" and "DMA controller descriptors" interchangeably.



The relevant language of claim 1 also requires that the claimed DMA descriptors are “used to fetch other POR sequencer descriptor fragments.” This concept is also disclosed in the ’084 Patent, as introduced above. In the exemplary illustration of Figure 5 and associated description, the “DMA controller descriptor 506 of the POR sequencer descriptor fragment 501, 502 points to the next POR sequencer descriptor fragment 502.” ’084 Pat., 6:44-46.

The loading of DMA controller descriptors by the POR Sequencer to the DMA controller's configuration register space during the boot-up process allows the associated DMA controller to be activated. As described, the process of "copying of system firmware 211 from the storage device 209 to the RAM 208" is illustrated in Figure 8, and "starts with the POR sequencer 207 reading a DMA controller descriptor 406 from the RAM 208 and then writing the DMA controller descriptor 406 to the configuration space of an available DMA controller 203 in block 801." '084 Pat., 7:50-56. "The POR sequencer 207 then activates the DMA controller 203 in block 802, which would then cause the DMA controller 203 to start the transfer of a system firmware fragment 211 from the storage device 209 to the internal RAM 208 via the storage device controller 204 in block 803." '084 Pat., 7:56-61. As such, the '084 Patent describes a DMA controller descriptor as being read by a POR sequencer 207, and loaded into an available DMA controller's configuration register space as register information to activate the associated DMA controller. Petitioner's characterization of "DMA descriptors" as being fully disclosed by Post's pointer overlooks this disclosure and fails to accord with the plain meaning of the term, "DMA descriptors."

Additionally, in Section IV of the Petition, which contains Petitioner's discussion of the technical background, Petitioner includes Section IV.C directed to Direct Memory Access ("DMA"). See Pet., 9-11. In that discussion, Petitioner

cites to a textbook entitled *Designing Embedded Hardware* to define a “descriptor” as ““a table specifying byte count, source address, destination address, and a pointer to the next descriptor.”” Pet., 11 (quoting Ex. 1008, 21).<sup>7</sup> Petitioner’s assertion that the claimed “DMA descriptor” is satisfied solely by Post’s pointer also contradicts Petitioner’s own evidence regarding what it contends must be included in a “descriptor.”

In summary, both the ’084 Patent and Petitioner’s own evidence establish that the plain and ordinary meaning of a DMA descriptor includes more than Post’s pointer.

Magnifying Petitioner’s error, Petitioner ignores that the asserted combination of Post and Kao does not include any DMA descriptors that could be included in Post’s boot code to be loaded. Petitioner relies on Kao to disclose a DMA engine used to load boot codes and controller codes stored in non-volatile memory. Pet., 42. But Petitioner does not and cannot identify any DMA controller descriptor in Kao. See Pet., 42-47. An independent search confirms that Kao fails to disclose any “descriptor,” much less a “DMA descriptor” as claimed. This is no surprise, as Kao has no need for any DMA descriptor to activate its DMA engine. In Kao, a NAND programmer 302 “burn[s] boot codes 132 (step

---

<sup>7</sup> The cited page for this definition—“21”—can be found on page 39 of the Ex. 1008 pdf submitted with the Petition.

304), controller codes 134 (step 306), and OS codes 136 (step 308) into the NAND flash memory 130” before booting, and “utiliz[es] the DMA engine located on the volatile memory chip to load boot and controller codes.” Kao, Fig. 3A (“BEFORE BOOTING”), 4:53-58, 3:3-4. This pre-loaded code for the DMA engine is what yields Kao’s alleged reduced complexity and cost of the host processor, and obviates any need for the loading of any DMA descriptors in Kao. *See* Kao, 2:64-3:9. Indeed, Kao’s purpose is to develop “an improved system and method for booting an embedded system” that does not include “initializing and managing DMA transfers ... as part of the boot sequence.” Kao, 2:5-15.

As such, whether considered alone or in combination, Post and Kao fail to disclose or suggest any “POR sequencer descriptor comprises ... Direct Memory Access (DMA) descriptors used to fetch other POR sequencer descriptor fragments” as presented in the asserted challenge. As discussed above, Kao fails to disclose any DMA descriptors to be loaded upon boot-up. Kao’s DMA engine code is preloaded into NAND flash memory prior to booting, is not stored in fragments, and does not need to be loaded to activate a DMA controller upon boot-up. As such, the combination of Post and Kao does not result in the incorporation of any DMA descriptors to the boot code pages of Post. Whether considered alone or in any combination, Post and Kao fail to disclose any DMA descriptors, much

less DMA descriptors used to fetch other POR sequencer descriptor fragments as claimed.

For all these reasons, Petitioner’s combination of Post, Kao, and Wilson fails to disclose or suggest at least “wherein the POR sequencer descriptor comprises ... Direct Memory Access (DMA) descriptors used to fetch other POR sequencer descriptor fragments,” as claimed.

**iii. Petitioner’s Proposed Combination of Post, Kao, and Wilson Also Fails to Disclose or Suggest the Steps of Verifying an Integrity of the POR Sequencer Descriptor and Detecting Any Error in the POR Sequencer Descriptor, as Claimed**

Claim 1 also recites, in relevant part, “verifying an integrity of the POR sequencer descriptor,” and “detecting any error in the POR sequencer descriptor.” The Petition fails to establish that these features are present or suggested by Petitioner’s proposed combination of Post, Kao, and Wilson. Here, Petitioner’s analysis fundamentally fails to account for the difference between a “POR sequencer descriptor” and “POR sequencer descriptor fragments” as expressly claimed in claim 1.

Petitioner alleges that “Post disclosed using next page pointers 302, prior page pointers 304, and boot data page indicators 306 to verify an integrity of the boot data.” Pet., 50 (citing Ex. 1003, ¶¶126-130). But this process of accounting for all boot data occurs with respect to fragments—pages—of boot code data in Post. Petitioner specifically quotes Post’s process for identifying the relevant

pages, stating “[T]he electronic device can use the next page pointers to verify the integrity of boot data storage.” Pet., 50 (quoting Post, 6:54-55). Post explains further, however, by stating 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.” Post, 7:21-23. Notably, Post is not verifying the integrity of boot code data that has been stored in or loaded from NVM, but instead is using the presence of pointers to ensure that all boot data pages—or fragments—are accounted for. Indeed, this is all Post proposes to achieve through its disclosed page-by-page process. As Post explains, “the loading of boot data can be accomplished by incrementally processing and validating pages as the pages are being read.” Post, 5:50-52. In this case, Post’s process of validating is limited to ensuring that all pages possessing pointers have been accounted for in sequential order, not whether the contents of those pages are accurate.

Thus, Post’s use of pointers is designed to ensure that every page of boot code data is accounted for in the nonvolatile memory, but does not verify the integrity of the boot code data on every page, and certainly does not verify the integrity of all boot code data “available in a form suitable for execution by control circuitry,” *i.e.*, once loaded to memory 220. Post, 5:48-50. Post teaches that the pointers “verify that pages expected to have boot data actually contain boot data (and vice versa).” Post, 6:22-24. These pointers may show that Post has identified

all boot code data pages, but these pointers do not ensure the accuracy or integrity of the data contained therein.

For the claimed “detecting any error in the POR sequencer descriptor” element, Petitioner identifies Post’s use of error correcting codes “on the contents of page 2.” Pet., 51 (citing Post, 6:54-7:5). But Post explains that this use of error correcting codes is to “confirm that page 2 provides boot data,” not to check the page for errors. Post, 6:58. Post states that the determination of whether a page contains boot data “can be performed by employing an error correcting code on the initial page data *to check for a valid codeword*, or by checking a program status of the initial page to determine whether the initial page is programmed or unprogrammed.” Post, 8:34-40 (emphasis added). Further, this process is performed on a page-by-page basis. Post discloses that the error correction occurs on page 2, which is in a block “in the boot data storage area of NVM 120,” or nonvolatile memory. Post, 6:21-24, 6:54-59. If this process results in “uncorrectable errors” on the page, Post can still find another copy of the boot data “elsewhere.” Post, 6:60-64. Thus, an error on a page does not represent an error in Post’s boot code data “available in a form suitable for execution by control circuitry.” Post, 5:48-50.

Finally, Post’s use of pointers to ensure that every page of boot code data is accounted for, and use of error correcting codes on individual pages of boot code

data to ensure those pages actually possess boot code, only address Post’s own boot code data. As discussed in the prior Section III.A.ii, Petitioner’s asserted combination fails to disclose every feature of the claimed “POR sequencer descriptor,” including at least “wherein the POR sequencer descriptor comprises ... Direct Memory Access (DMA) descriptors used to fetch other POR sequencer descriptor fragments.” Petitioner’s combination fails to disclose such DMA descriptors as part of any POR sequencer descriptor, and therefore necessarily fails to disclose any verification of integrity or error correction of any DMA descriptors as part of a POR sequencer descriptor, as required by the challenged claim.

For all these reasons, Petitioner’s combination of Post, Kao, and Wilson fails to disclose or suggest at least “verifying an integrity of the POR sequencer descriptor” and “detecting any error in the POR sequencer descriptor,” as claimed.

**B. Ground 1’s Challenge Fails to Establish the Unpatentability of Claim 1**

For all of the reasons presented above, Petitioner’s proposed combination of Post, Kao, and Wilson fails to disclose or suggest every claimed feature of claim 1. Therefore, claim 1 survives Petitioner’s Ground 1 challenge. Further, the remaining challenged claims either depend from claim 1 or recite features substantially similar to claim 1. *See, e.g.,* Pet., 59-61 (relying on the analysis of claim 1 to challenge claim 19’s “article of manufacture”). Therefore, for the same reasons already present in this Section III, all challenged claims, including

independent claim 19, along with dependent claims 5, 8, 9, and 20, survive Petitioner's challenge.

#### **IV. CONCLUSION**

For the reasons presented above, the Petition's Ground is deficient and no *inter partes* review should be instituted.

Dated: August 15, 2023

/s/ Wayne M. Helge

Wayne M. Helge (Reg. No. 56,905)

NOSSAMAN, LLP

1401 New York Ave, NW

Suite 800

Washington, DC 20005

Telephone: (202) 887-1400

Email: whelge@nossaman.com

James T. Wilson (Reg. No. 41,439)

DAVIDSON BERQUIST JACKSON &

GOWDEY, LLP

8300 Greensboro Dr., Suite 500

McLean, VA 22102

Telephone: (571) 765-7700

Fax: (571) 765-7200

Email: jwilson@dbjg.com

Counsel for Patent Owner

**CERTIFICATE OF WORD COUNT**

The undersigned certifies that the foregoing PATENT OWNER'S PRELIMINARY RESPONSE complies with the type-volume limitation in 37 C.F.R. § 42.24(b)(1). According to the word-processing system's word count, the brief contains 4,986 words, excluding the parts of the brief exempted by 37 C.F.R. § 42.24(a).

By: /s/ Wayne M. Helge  
Wayne M. Helge (Reg. No. 56,905)  
Attorney for Patent Owner

**CERTIFICATE OF SERVICE**

I hereby certify that on August 15, 2023, a true and correct copy of the foregoing document was served via email, by consent, to Petitioner by serving the correspondence email addresses of record as follows:

Christina McCullough  
Reg. No. 58,720  
PERKINS COIE LLP  
1201 Third Avenue  
Suite 4900  
Seattle, Washington 98101-3099  
(206) 359-6897  
mccullough-ptab@perkinscoie.com  
Intel-BitMicro-  
084IPRService@perkinscoie.com

Lori A. Gordon  
Reg. No. 50,633  
PERKINS COIE LLP  
700 Thirteenth Street, N.W.,  
Suite 800  
Washington, DC 20005-3960  
(202) 654-6200  
gordon-ptab@perkinscoie.com

Christopher Kelley  
Reg. No. 42,714  
PERKINS COIE LLP  
3150 Porter Drive  
Palo Alto, CA 94304-1212  
(650) 838-4300  
kelley-ptab@perkinscoie.com

Chad S. Campbell  
PERKINS COIE LLP  
2901 N. Central Avenue  
Suite 2000  
Phoenix, AZ 85012  
campbell-ptab@perkinscoie.com

/s/ Wayne M. Helge  
Wayne M. Helge (Reg. No. 56,905)  
Attorney for Patent Owner