

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

INTEL CORPORATION,
Petitioner,

v.

VLSI TECHNOLOGY LLC,
Patent Owner.

IPR2019-01192
Patent 7,523,331 B2

Before BART A. GERSTENBLITH, MINN CHUNG, and
KIMBERLY McGRAW, *Administrative Patent Judges*.

McGRAW, *Administrative Patent Judge*.

JUDGMENT
Final Written Decision
Determining Challenged Claim Unpatentable
35 U.S.C. § 318(a)

I. INTRODUCTION

In this *inter partes* review, instituted pursuant to 35 U.S.C. § 314, Intel Corporation (“Petitioner”) challenges claim 7 of U.S. Patent No. 7,523,331 B2 (Ex. 1001, “the ’331 patent”), owned by VLSI Technologies Corp. (“Patent Owner”). This Final Written Decision is entered pursuant to 35 U.S.C. § 318(a) and 37 C.F.R. § 42.73. For the reasons discussed below, Petitioner has shown by a preponderance of the evidence that claim 7 of the ’331 patent is unpatentable.

A. *Procedural History*

Petitioner filed a Petition for *inter partes* review of claim 7 of the ’331 patent. Paper 3 (“Pet.”). Patent Owner filed a Preliminary Response. Paper 9. Petitioner then filed an authorized Reply to address Patent Owner’s arguments regarding 35 U.S.C. §§ 314(a), 325(d) (Paper 11), to which Patent Owner filed an authorized Sur-reply (Paper 12). Applying the standard set forth in 35 U.S.C. § 314(a), which requires demonstration of a reasonable likelihood that Petitioner would prevail with respect to at least one challenged claim, we instituted an *inter partes* review of the challenged claim. Paper 15 (“Inst. Dec.”).

Following institution, Patent Owner filed a Patent Owner Response (Paper 21, “PO Resp.”), Petitioner filed a Reply (Paper 26, “Pet. Reply”), and Patent Owner filed a Sur-reply (Paper 35, “PO Sur-reply”). Patent Owner’s request for authorization to file a motion to strike portions of Petitioner’s Reply and corresponding paragraphs of the Reply Declaration of Dr. Carl Sechen was denied. Paper 32 (Order). Patent Owner also requested, and was denied, authorization to file a belated request for rehearing our Decision on Institution or a motion to dismiss the proceeding

in view of the August 18, 2020, USPTO guidance¹ regarding how the office will treat a petitioner's reliance on statements made in the specification of a challenged patent. Paper 36 (Order). The parties then filed authorized briefing to address Petitioner's reliance on statements in the specification of '331 patent (Ex. 1001, 1:6–9, 1:34–43, "Applicants Admitted Prior Art" or "AAPA"). Paper 39 (Petitioner's Supplemental Briefing on AAPA); Paper 41 (Patent Owner's Brief on Petitioner's Reliance on AAPA). A prehearing conference was held on September 25, 2020, during which Petitioner's request to strike portions of Exhibit 2013 was denied. Ex. 1035 (Transcript of September 25, 2020, Prehearing Conference), 20:8–22. An oral hearing was held on October 7, 2020, and a copy of the hearing transcript has been entered into the record. Paper 45 ("Tr.").

B. Real Parties-in-Interest

Petitioner identifies Intel Corporation as the real party-in-interest for Petitioner. Pet. 1. Patent Owner identifies VLSI Technology LLC and CF VLSI Holdings LLC as the real parties-in-interest for Patent Owner. Paper 7, 1. *See* 37 C.F.R. § 42.8(b)(1).

C. Related Matters

The parties represent that the '331 patent is at issue in *VLSI Technology LLC v. Intel Corp.*, No. 18-966 (D. Del.). Pet. 1; Prelim. Resp. 9. In addition to the '331 patent, there are four other patents also at issue in the co-pending litigation. *See* Ex. 1029 (stating claims of U.S.

¹ The August 18, 2020, USPTO Memorandum addressing Treatment of Statements of the Applicant in the Challenged Patent in *Inter Partes* Reviews under § 311 ("Guidance Memo") is available at https://www.uspto.gov/sites/default/files/documents/signed_aapa_guidance_memo.pdf.

Patent Nos. 6,612,633, 7,246,027, 7,247,552, and 8,081,026 are also asserted in *VLSI Technology LLC v. Intel Corp.*, No. 18-966 (D. Del.)).

D. The '331 Patent

The '331 patent, titled “Power Saving Operation of an Apparatus with a Cache Memory,” is directed to a method of reducing power consumption by a data processing apparatus. Ex. 1001, code (54), 1:6–7. The '331 patent explains that, conventionally, cache memory is one of the circuits that are deactivated when an apparatus is switched to low power mode because cache merely stores redundant copies of part of the data and/or instructions that were used during previous processing. *Id.* at 1:45–49. The invention, however, keeps at least a part of a cache memory active in the low power mode while the main memory is deactivated, or at least not fully operational. *Id.* at 1:54–59. The '331 patent explains that before switching to the low power mode, a program of instructions that are needed to perform a function while the apparatus is in the low power mode can be loaded into the cache memory for later use. *Id.* at 1:59–63. When an instruction processing circuit performs the function in the low power mode, it loads the instructions from cache memory without activating the main memory. *Id.* at 1:63–66. Because the main memory is generally larger than cache memory, main memory generally consumes more power than cache memory; power consumption is reduced by using the relatively smaller cache memory as only memory in the low power operating mode. *Id.* at 1:66–2:6.

Figure 1, reproduced below, illustrates an apparatus that supports a low power operating mode. *Id.* at 2:10–11.

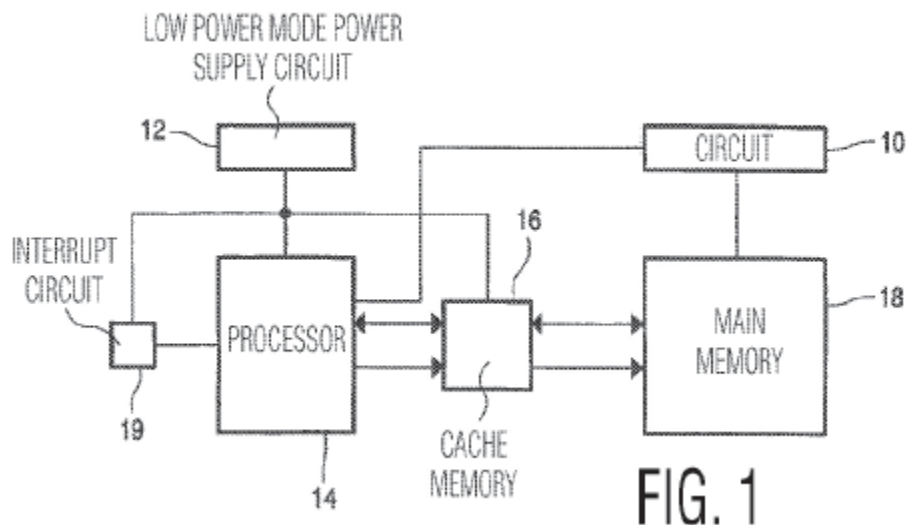


Figure 1 above illustrates an apparatus with processor 14, cache memory 16, and main memory 18. *Id.* at 2:13–16. The '331 patent explains that in a “normal operating mode cache memory 16 functions as a conventional cache memory 16.” *Id.* at 2:35–36. When processor 14 detects that it is no longer necessary to operate in the normal mode, processor 14 executes a “switchover program” to switch over to low power. *Id.* at 3:10–12. As part of the switch, the switchover program causes processor 14 to signal to cache memory 16 to load all instructions of an interrupt program from main memory into cache memory. *Id.* at 3:16–19. The interrupt program is stored at addresses in main memory that have been selected so that all instructions of the interrupt program can be stored together in cache memory 16. *Id.* at 3:19–22. The '331 patent states that, in one example, the switchover program is provided with the start address and the end address of the interrupt program, so that the switchover program can address all memory locations from the start address to the end address, forcing them to be loaded into cache memory 16. *Id.* at 3:24–28. If it is known that main memory 18

returns lines of instruction and/or data, only one address from each line needs to be addressed. *Id.* at 3:28–30. In another example, where cache memory is provided with a conventional locking mechanism that enables processor 14 to signal that certain instructions or data must be kept in cache, the switchover program can make processor 14 signal to cache memory that the instructions of the interrupt program must be kept. *Id.* at 3:32–38.

Alternatively, the interrupt program and the switchover program may be “stored at addresses that are selected so that it is ensured in advance that the switchover program can load the entire interrupt program into cache memory 16 without being subsequently discarded.” *Id.* at 3:38–43. In yet another alternative, processor 14 may “contain[] an instruction to cause specified instructions, or specifically the interrupt routine, to be loaded into cache memory 16.” *Id.* at 3:43–46.

After the instructions needed to perform a function are stored in cache memory 16 and the apparatus enters the low power operating mode, the interrupt program processor can perform the function in the low power mode by loading the instructions from cache memory and execute the interrupt program without recourse to main memory 18. *See id.* at 1:63–66, 3:65–67.

E. Challenged Claim

Claim 7, the only claim challenged in this proceeding, is reproduced below with formatting and bracketing added for clarity.

7. [a] A method of operating an apparatus that contains an instruction processing circuit, a main memory addressable by the instruction processing circuit and a cache memory, the method comprising:
 - [b] using the cache memory and the main memory in a normal operating mode,
 - [c] to cache in cache memory a part of data and/or instructions that the instruction processing circuit addresses in the main memory

during execution, and to substitute cached data and/or instructions when the instruction processing circuit addresses the data and/or instructions in the main memory;

[d] storing, in the main memory, a program of instructions for executing an interrupt function during operating in a low power operating mode,

[e] wherein the interrupt program is stored at addresses in main memory that have been selected so that all instructions of the interrupt program can be stored together in the cache memory;

[f] detecting that it is no longer necessary to operate in the normal operating mode;

[g/h] switching to the low power operating mode once it is detected that it is no longer necessary to operate in the normal operating mode, by

[g] loading the interrupt program into the cache memory from the main memory, wherein all instructions of the interrupt program are stored together in the cache memory;

[h] deactivating the main memory to reduce power consumption, but keeping active at least a part of the cache memory, that is needed for retrieving the interrupt program and for executing the interrupt function;

[i] executing the interrupt program from said at least part of the cache memory.

F. Prior Art and Asserted Grounds

Petitioner asserts that claim 7 is unpatentable under 35 U.S.C. § 103(a) on the following grounds (Pet. 5):

Claim Challenged	35 U.S.C. §	Basis
7	103	Irie, ² AAPA ³
7	103	Irie, AAPA, Bourekas ⁴

G. Testimonial Evidence

Petitioner relies on a Declaration (Ex. 1008) and a Reply Declaration (Ex. 1032) by Carl Sechen, Ph.D. Patent Owner cross-examined Dr. Sechen via deposition. *See* Ex. 2012 (First Deposition of Dr. Sechen); Ex. 2020 (Second Deposition of Dr. Sechen). Patent Owner relies on a declaration by Glenn Reinmann, Ph.D. (Ex. 2013). Petitioner cross-examined Dr. Reinmann via deposition. *See* Ex. 1033.

II. DISCUSSION

A. Legal Principles

To prevail on its challenge to Patent Owner’s claims, Petitioner must demonstrate by a preponderance of the evidence that the claim is unpatentable. 35 U.S.C. § 316(e); 37 C.F.R. § 42.1(d). The petitioner “has the burden from the onset to show with particularity why the patent it challenges is unpatentable.” *Harmonic Inc. v. Avid Tech., Inc.*, 815 F.3d 1356, 1363 (Fed. Cir. 2016) (citing 35 U.S.C. § 312(a)(3) (requiring *inter*

² JP 2000-105639 A, published Apr. 11, 2000 (Ex. 1003; Ex. 1002 (English translation), “Irie”). Unless otherwise indicated, citations to foreign language documents will be made to the English translation of the document.

³ Petitioner asserts the material at Exhibit 1001, 1:6–9, 1:34–43, constitutes Applicant Admitted Prior Art (Ex. 1001, “AAPA”). *See* Pet. 1; *see also* Pet. 30–31, 35 (citing Ex. 1001, 1:34–43).

⁴ U.S. Patent No. 5,694,567, issued Dec. 2, 1997 (Ex. 1005, “Bourekas”).

partes review petitions to identify “with particularity . . . the evidence that supports the grounds for the challenge to each claim”). Apart from limited circumstances not presented here, this burden does not shift to Patent Owner. See *Dynamic Drinkware, LLC v. Nat’l Graphics, Inc.*, 800 F.3d 1375, 1378 (Fed. Cir. 2015) (citing *Tech. Licensing Corp. v. Videotek, Inc.*, 545 F.3d 1316, 1326–27 (Fed. Cir. 2008)) (discussing the burden of proof in *inter partes* review).

A claim is unpatentable for obviousness under 35 U.S.C. § 103(a) 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) when in evidence, objective indicia of non-obviousness (i.e., secondary considerations⁵). *Graham v. John Deere Co.*, 383 U.S. 1, 17–18 (1966).

B. Claim Construction

For petitions filed on or after November 13, 2018, such as here, we apply the same claim construction standard that would be used to construe the claim in a civil action under 35 U.S.C. § 282(b), including construing the claim in accordance with the ordinary and customary meaning of such claim, as understood by one of ordinary skill in the art, and the prosecution history

⁵ Patent Owner does not present arguments or evidence of such secondary considerations; therefore, secondary considerations do not constitute part of our analysis.

pertaining to the patent. 37 C.F.R. § 42.100(b) (2019); *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc).

We address the disputed claim terms discussed by the parties.

1. “*addresses in main memory that have been selected so that all instructions of the interrupt program can be stored together in the cache memory*”

Petitioner asserts that a POSITA would have understood “addresses in main memory that have been selected *so that all instructions of the interrupt program can be stored together in the cache memory*” recited in claim element 7[e] to at least encompass “addresses in main memory that have been selected so that all instructions of the interrupt program can be stored in [1] a *single cache memory* or a [2] *single part of cache memory* at the same time.” Pet. 21 (emphasis altered); *see also* Ex. 1001, 8:22–24 (claim element 7[g] similarly reciting “wherein *all instructions of the interrupt program are stored together in the cache memory*” (emphasis added)).

Petitioner contends its proposed construction is supported by the ’331 patent’s description that all instructions of the interrupt program must be present in the cache memory during the low power operating mode so the interrupt program can be executed from the cache memory. *Id.* at 21–22 (citing Ex. 1001, 3:48–67, 3:55–61, 3:64–67; Ex. 1008 ¶ 42). In its Preliminary Response, Patent Owner did not provide an explicit construction but appeared to contend “stored together in the cache memory” means stored in contiguous locations in the cache memory. *See* Prelim. Resp. 27 (stating Irie does not disclose selecting “addresses in main memory for storage of its interrupt program so that the program can later be stored in *contiguous addresses in cache*” (emphasis added)).

In our Decision on Institution, we determined that the term “together” in the phrase “addresses in main memory that have been selected so that all instructions of the interrupt program can be stored together in the cache memory” recited in claim element 7[e] does not require storing the instructions of the interrupt program in *contiguous* locations in a single cache, as argued by Patent Owner, but does require more than merely storing the instructions in a single cache, as argued by Petitioner. Dec. Inst. 16–17. We further stated the disputed phrase “encompass[es] storage of all instructions of an interrupt program *in part of a cache memory*.” *Id.* at 25 (emphasis added); *see also id.* at 16 (discussing the ’331 patent’s teaching that instructions for the interrupt program must be present in the cache memory that remains active during the low power operating mode so that the interrupt program can be executed from the cache memory).

Petitioner does not dispute our construction. Pet. Reply 9 (stating “the Board correctly found the term ‘together’ does not require storing the instructions of the interrupt program in contiguous locations in a single cache, but does require more than merely storing the instructions in a single cache”); *but see* Pet. Reply 17 n.12 (stating “[a]lternatively, interpreting ‘together’ as encompassing ‘in a single cache’ (instead of ‘in a part of’) is far more consistent with the plain meaning of ‘together’ than Patent Owner’s artificially narrow construction”) (citing Ex. 1032 ¶ 46 n.2).

In its Patent Owner Response, Patent Owner disagrees with our construction and again asserts that the claim term “stored together” in the phrase “all instructions of the interrupt program are stored together in cache memory” should be construed as “stored contiguously.” PO Resp. 18.

Having reviewed the parties’ arguments, based on a full record, we see no reason to change our construction and determine that “addresses in

main memory that have been selected so that all instructions of the interrupt program can be stored together in the cache memory” recited in claim element 7[e] does not require storing the instructions of the interrupt program in contiguous locations in a single cache, but does require more than merely storing the instructions in a single cache, such as storing the instructions in a part of cache (e.g., a part of the cache that is not deactivated when another part of the cache is deactivated).

As stated in our Decision to Institute, we agree with Patent Owner that this disputed phrase requires something more than merely storing the recited instructions in a *single cache memory*. Dec. Inst. 15. The claim does not recite “wherein all instructions of the interrupt program are stored in the cache memory” but rather, requires that the instructions “are stored *together* in the cache memory.” Ex. 1001, 8:5–24 (emphasis added). A construction that merely requires the instructions to be stored in the cache memory would render the term “together” superfluous. *See Merck & Co. v. Teva Pharm. USA, Inc.*, 395 F.3d 1364, 1372 (Fed. Cir. 2005) (rejecting a proposed claim construction that would render claim terms superfluous).

We disagree with Patent Owner’s contention, however, that stored *together* in the cache memory requires storage in *contiguous locations* in the cache. The plain language of the claim does not require “contiguous” storage. Nor does the Specification state that the stored instructions must be contiguous. *See* Ex. 1001. Patent Owner asserts that the ’331 patent teaches that “excess storage capacity of cache memory 16, which is not needed during operation in the low power operating mode may be deactivated.” PO Resp. 20 (quoting Ex. 1001, 4:47–49). Patent Owner contends that storing the claimed program of instructions contiguously best achieves the ’331 patent’s objective by maximizing the amount of excess storage capacity

of cache memory that may be deactivated in order to achieve this benefit. *Id.* (citing Ex. 2013 ¶ 47).

The '331 patent, however, does not state that the amount of cache to be deactivated during low power mode must be maximized. Nor does the '331 patent state anywhere that instructions should be stored contiguously. Simply put, there is no indication in the '331 patent that instructions *must* be stored contiguously, that the amount of excess storage capacity of cache memory that may be deactivated *must* be maximized, or that storing instructions contiguously would achieve any such goal.

Rather, the Specification of the '331 patent states that “*part of cache memory 16* may be deactivated in the low power operating mode in order to reduce power supply consumption.” Ex. 1001, 4:34–36 (emphasis added); *see also, e.g., id.* at 2:57–62, 4:34–49 (describing different parts of cache memory and how these different parts can be deactivated or remain active during the low power operating mode). The Specification further states that the instructions of an interrupt program “can be stored together in cache memory 16” (*id.* at 3:19–22) and that it is assumed that “cache memory 16 will retain all addressed data” (*id.* at 3:31–32). Additionally, the Specification explains that when an apparatus has entered a low power operating mode in response to some event, such as the actuation of a user interface button, the interrupt program processor can “address[] the instructions of the interrupt program that are stored in cache memory 16 and is therefore able to execute the interrupt program without recourse” to the main memory. *Id.* at 3:55–60, 3:64–67. Thus, the Specification supports our understanding that the program of instructions that are stored *together* in cache memory are stored in the part of cache memory that remains active in

low power operating mode when other parts of cache memory are deactivated in the low power operating mode.

We also credit the persuasive testimony of Dr. Sechen that program instructions that are stored with or without intervening instructions (i.e., including instructions that are not stored contiguously) are stored together when the cache has used and unused portions. Ex. 2012, 106:21–24, 114:13–115:1; Ex. 1032 ¶ 43.

Patent Owner cites to various English language dictionaries that purportedly define “together” in the relevant definition as “continuously” or “uninterruptedly” to supports its contention that the plain and ordinary meaning of “together” means “contiguous.” PO Resp. 19–20 (stating Ex. 2014 [Webster’s New World Dictionary] defines “together” as “continuously”; Ex. 2015 [Encarta World English Dictionary] defines “together” as “Uninterruptedly”); Ex. 2016 [Oxford Dictionary] defines “together” as Uninterruptedly; Ex. 2017 [Chambers Dictionary] defines “together” as “continuously” and “so as to be in contact, joined or united”). Each of these dictionaries, however, also provides a definition of “together” that does not require “continuously,” “uninterruptedly,” or “contiguous.” *See, e.g.*, Ex. 2014, 5 (in or into one gathering, group, mass, or place); Ex. 2015, 1872 (“in company with others in a group or in a place); Ex. 2016, 769 (“with or near to another”); Ex. 2017, 1595 (“with someone or something else, in company; in or into one place”). Thus, the cited dictionary definitions do not change our understanding of the plain and ordinary meaning of the disputed term.

Patent Owner also contends construing storing “together” as storing in a “part of a cache” would render the term superfluous and meaningless because a “program stored in a cache memory is essentially *always* stored in

a *part* of the cache memory.” PO Resp. 25 (emphasis altered) (citing Ex. 2013 ¶ 57); *see also id.* at 25–28 (arguing that “storing together” requires more than “storing in a part of cache”). However, as explained above, our construction encompasses storing a program of instructions in a “part of cache” that remains active in low power operating mode when other parts of cache memory are deactivated in the low power operating mode. Additionally, in systems with multiple cache memories, requiring that the “instructions of the interrupt program can be stored together in the cache memory” would require that the instructions be stored “together” in just one of those cache memories. Thus, we disagree that a construction of storing “together” as storing in a “part of a cache” renders “together” superfluous or meaningless.

Patent Owner also contends Petitioner’s expert, Dr. Sechen, admitted that the Board’s construction would render “together” superfluous. *Id.* at 26–28 (citing Ex. 2012, 102:6–13, 105:13–17). A review of the full testimony, however, shows Dr. Sechen testified that program instructions stored with or without intervening instructions are stored “together” when the cache has *used and unused* portions. *See* Ex. 2012, 106:21–24, 114:13–115:1.

Our construction is also consistent with the prosecution history. As explained by Patent Owner, “the Examiner allowed the claim over art [i.e., Ex. 1022, “Hamilton”⁶], that discloses simply storing in ‘*a single cache memory.*’” PO Resp. 22 (emphasis added). To overcome a § 102 rejection (*see* Ex. 1012, 2) based on Hamilton, Patent Owner amended the predecessor to claim 7 to add the “storing together” language that ultimately

⁶ U.S. Patent No. 7,058,829 B2.

became claim elements 7[d], 7[e], and 7[g]. *See* Ex. 1013, 3–4; *see also id.* at 4 (stating the claim has been amended to include “storing, in the main memory, a program of instructions for executing a function during operating in a low power operating mode, wherein the program is stored at addresses in main memory that have been selected so that all instructions of the interrupt program can be stored together in the cache memory” and to particularly point out that “all instructions of the program are stored together in the cache memory”). Patent Owner did not contend that the amended claim language required *contiguous* storage as opposed to storage in a particular *part of cache*. *See* Ex. 1014, 3.

Therefore, for the foregoing reasons, based on the complete record, we maintain our determination that the term “together” in the phrase “addresses in main memory that have been selected so that all instructions of the interrupt program can be stored together in the cache memory” recited in claim element 7[e] does not require storing the instructions of the interrupt program in contiguous locations in a single cache, but does require more than merely storing the instructions in a single cache, such as storing the instructions in a part of cache (e.g., a part of the cache that is not deactivated when another part of the cache is deactivated).

2. “*data and/or instructions that the instruction processing circuit addresses in the main memory*” and “*when the instruction processing circuit addresses the data and/or instructions in the main memory*”

Petitioner asserts that in the co-pending district court litigation, the parties each proposed a different construction of two phrases recited in claim element 7[c], namely (1) “data and/or instructions that the instruction processing circuit addresses in the main memory” and (2) “when the instruction processing circuit addresses the data and/or instructions in the

main memory.” Petitioner states that in the district court proceeding, it proposed construing the first phrase (“data and/or instructions that the instruction processing circuit addresses in the main memory”) as:

data and/or instructions whose main memory address the instruction processing circuit outputs on its address/data interface,

(Pet. 32 n.10 (citing Ex. 1006, Ex. A, 3) while Patent Owner argued that the phrase means:

data and/or instructions that the instruction processing circuit identifies using addresses with which they can be retrieved from main memory

(*id.* at 34 n.13 (citing Ex. 1006, Ex. A, 3)). Petitioner also states that in the district court proceeding it argued the second phrase (“when the instruction processing circuit addresses the data and/or instructions in the main memory”) means:

when the instruction processing circuit outputs the main memory address of the data and/or instructions on its address/data interface

(*id.* at 32 n.10 (citing Ex. 1006, Ex. A, 3) while Patent Owner proposed that the phrase means:

when the instruction processing circuit identifies the data and/or instructions using addresses with which they can be retrieved from main memory.

(*id.* at 34 n.13 (citing Ex. 1006, Ex. A, 3)).

Petitioner also asserts that it is not necessary to construe the phrases recited in claim element 7[c] as Irie teaches the recited subject matter under each construction. *Id.* at 31–32 (citing Ex. 1008 ¶ 78).

In our Decision on Institution, we determined that neither term required construction to resolve any issues in dispute. Dec. Inst. 18. Neither party challenges this determination. *See generally* PO Resp.; Pet. Reply; PO

Sur-reply. Based upon a full record, we maintain our determination that neither of the identified phrases recited in claim 7[c] requires an express construction in order to resolve issues in dispute.

C. Level of Ordinary Skill in the Art

In determining whether an invention would have been obvious at the time it was made, we consider the level of ordinary skill in the pertinent art at the time of the invention. *Graham*, 383 U.S. at 17. Petitioner contends a person of ordinary skill in the art at the time of the alleged invention (“POSITA”) would have had at least: (1) an undergraduate degree in electrical engineering (or an equivalent subject), together with at least two years of postgraduate experience designing cache systems; or (2) a master’s degree in electrical engineering (or equivalent subject) together with at least one year of post-graduate experience in designing cache systems. Pet. 22 (citing Ex. 1008 ¶ 33). Patent Owner did not articulate a level of skill for a POSITA in its Preliminary Response or Patent Owner Response.

In our Institution Decision, we adopted Petitioner’s definition of a person of ordinary skill in the art at the time of the claimed invention, except that we deleted the qualifier “at least” to eliminate vagueness as to the appropriate level of education. The qualifier expands the range without an upper bound, i.e., encompassing a Ph.D. degree and beyond, and thus does not meaningfully indicate the level of ordinary skill in the art.

Neither party disputes our definition of the level of ordinary skill in the art. We see no reason to change our definition based on the complete record and, thus, maintain our definition for the purposes of this Decision.

D. Asserted Unpatentability of Claim 7 over Irie in view of AAPA

Petitioner contends that claim 7 would have been obvious under 35 U.S.C. § 103(a) over Irie in view of AAPA. Pet. 23–54; Pet. Reply 1–22.

Patent Owner opposes asserting that (1) Irie does not teach executing an interrupt function during operating in a low power mode, as required by claim element 7[d], (2) Irie does not teach that instructions for executing an interrupt function are “stored together in cache memory” as required by claim elements 7[e] and 7[g], and (3) Petitioner improperly relies upon AAPA. PO Resp. 1–39; PO Sur-reply 1–17; Paper 41.

For the reasons that follow, we determine Petitioner has shown by a preponderance of the evidence that claim 7 is unpatentable over Irie in view of AAPA.

1. Overview of Irie

Irie, titled “Energy-Saving Circuit,” discloses a mobile information terminal having CPU 10, ROM 21, RAM 22, and cache memory 11 having areas 11A and 11B. *See* Ex. 1002, code (54), ¶¶ 10–13. The CPU has an energy-savings mechanism that allows operation in “normal mode” and “energy-savings mode.” *Id.* ¶ 11. When the terminal is in “normal operating mode,” cache memory 11 is set to a first operating mode, in which both areas 11A and 11B operate as cache memory. *Id.* ¶ 23. In a second operating mode, cache area 11A is set as cache memory and area 11B is set as general-purpose memory. *Id.* ¶¶ 12, 23. When switching CPU 10 from a normal operating mode to an energy-saving operating mode, the CPU executes a first program in an external ROM 21, in order to copy a second program to internal cache memory 11, and thereafter executes the second program, which has been copied into memory 11. *Id.* at code (57). By doing so, the CPU turns the power to ROM 21 off and enters the energy savings operating mode. *Id.* When switching CPU 10 from the energy-saving operating mode back to the normal operating mode, CPU 10 executes the second program that has been copied into memory 11, turns the power to

ROM 21 back on, and connects ROM 21 to CPU 10, thereafter switching to execution of the first program and returning to the normal operating mode.

See id.

2. *AAPA*

Petitioner relies upon the following statements in the '331 patent to support its contention that claim 7 is unpatentable:

Known apparatuses that contain a computer processor and a main memory with data and/or instructions for use by the processor are often provided with a cache memory in order to speed up execution. The cache memory temporarily stores copies of part of the data and/or instructions that the processor has addressed in main memory, so that it can be retrieved given its main memory address. When the processor addresses such data and/or instructions again, the cache memory substitutes the cached data and/or instructions for the data and/or instructions from main memory. . . . Of course, cache memory is conventionally one of the circuits that are deactivated when the apparatus is switched to the low power mode, because it merely stores redundant copies of part of the data and/or instructions that were used during previous processing.

Ex. 1001, 1:34–49; *see also, e.g.*, Pet. 30–31 (citing Ex. 1001, 1:34–49 to support its argument that the cited art teaches the subject matter of claim element 7[c]).

Petitioner also points to statements in the '331 patent that, in disclosed embodiments, cache memory 16 functions as *conventional cache memory* when in the normal operating mode, such as:

In the normal operating mode cache memory 16 functions as a *conventional cache memory* 16. When processor 14 needs an instruction and/or data, processor 14 outputs the address with which the instruction and/or data is addressed in main memory on its address/data interface. Cache memory 16 receives the address and tests whether an instruction and/or data corresponding to the address is stored in cache memory 16.

Techniques for this type of testing *are known per se*. If the addressed instruction and/or data is available in cache memory 16, cache memory 16 returns the instruction and/or data from cache memory 16. If the addressed instruction and/or data is not available in cache memory 16, cache memory forwards the address to main memory 18, which returns the instruction and/or data to cache memory 16. Cache memory 16 then supplies the instruction and/or data from main memory 18 to processor 14. Preferably, cache memory 16 also stores a copy of this instruction and/or data, for later use, when processor 14 uses the address again.

Id. at 2:35–52 (emphases added); *see also, e.g.*, Pet. 30 (citing Ex. 1001, 2:35–52).

3. *Analysis of Claim 7*

a) *Element 7[a] (Preamble)*

The preamble of claim 7 recites a “method of operating an apparatus that contains an instruction processing circuit, a main memory addressable by the instruction processing circuit and a cache memory, the method comprising.” Ex. 1001, 7:39–42. Petitioner asserts that Irie discloses an apparatus (mobile information terminal) that contains an instruction processing circuit (CPU 10), a main memory (ROM 21)⁷ addressable by the instruction processing circuit (CPU 10), and cache memory (cache 11). *See* Pet. 23–27 (citing, *inter alia*, Ex. 1002 ¶¶ 10–12, 28, 55–56, Fig. 1; Ex. 1008 ¶¶ 66–67). Petitioner also asserts that Irie executes program routine 300 to

⁷ Petitioner also asserts that Irie’s RAM 22 is a main memory. *See* Pet. 24–30, 36, 37, 40–44, 49–51, 55, 60–61. Petitioner, however, only relies on ROM 21, and not RAM 22, as satisfying the limitations of claim element 7[g] that require that the interrupt program is stored at addresses in the main memory that have been selected so that all instructions of the interrupt program can be stored together in the cache memory. *See* Pet. 41–44; *see also id.* at 44 (stating the selected addresses of the main memory are the starting, final, and intervening addresses in ROM 21).

copy instruction content for a program from ROM 21 to cache memory 11. *Id.* at 25 (citing Ex. 1002 ¶ 28). For example, Petitioner asserts Irie's routine 300 is executed to determine a starting address in ROM 21 of instruction content for a program; CPU 10 reads the instruction content stored at this starting address and copies it to a temporary location in RAM 22; instructions are then moved from the temporary location in RAM 22 to cache memory 11; CPU 10 then determines a subsequent address location in ROM 21 where further program instruction content is stored, reads the stored content and copies it to the same temporary location in RAM 22; the instructions are then moved from the temporary location in RAM 22 to cache memory 11; and the process is continued until a final address holding instruction content in ROM 21 is reached. *Id.* (citing Ex. 1002 ¶¶ 27–31). Petitioner asserts that because CPU 10 determines the address locations of ROM 21 in this process in order to read the instruction content stored at these address locations, the main memory (ROM 21) is addressable by the instruction processing circuit (CPU 10). *Id.* (citing Ex. 1002 ¶¶ 27–31; Ex. 1008 ¶¶ 61, 64).

We agree with Petitioner's analysis and find that Irie teaches or suggests the subject matter recited in the preamble of claim 7 for the reasons explained by Petitioner.⁸ *See* Pet. 23–28; Ex. 1008 ¶¶ 55–68. Patent Owner has not raised arguments against the preamble in its Patent Owner Response; therefore any such arguments are waived. *See* Paper 16 (Scheduling Order), 7 (stating that any arguments for patentability not raised in the response may be deemed waived).

⁸ Because we determine Petitioner has sufficiently shown the cited art teaches the subject matter recited in the preamble, we need not determine whether the preamble is limiting.

b) Element 7[b]

Claim element 7[b] recites “using the cache memory and the main memory in a normal operating mode.” Ex. 1001, 8:1–2. Petitioner asserts Irie’s disclosure of operating CPU 10 and using cache memory (cache area 11A and cache area 11B of cache memory 11) and main memory (ROM 21) in Irie’s “normal operating mode” satisfies this claim element. *See* Pet. 28–30 (citing Ex. 1002 ¶¶ 11–14, 22–25, 37–39, 41; Ex. 1008 ¶¶ 69–72).

We agree with Petitioner’s analysis and find that, for the reasons explained by Petitioner, Irie discloses using cache memory and main memory in a normal operating mode as required by claim element 7[b]. *See* Pet. 28–30; Ex. 1008 ¶¶ 69–72. For example, Irie states “CPU 10 [has] an energy-saving mechanism which allows operation in normal mode.” Ex. 1002 ¶ 11; Pet. 28. Irie also states that when “the mobile information terminal is in normal operating mode . . . cache memory 11 is set to a first operating mode, in which both areas 11A and 11B operate as cache memory.” Ex. 1002 ¶ 23. Patent Owner has not raised arguments against this limitation in its Patent Owner Response; therefore, any such arguments are waived.

c) Element 7[c]

Claim element 7[c] requires “using the cache memory and main memory in a normal operating mode” recited in claim element 7[b] “to cache in cache memory a part of data and/or instructions that the instruction processing circuit addresses in the main memory during execution, and to substitute cached data and/or instructions when the instruction processing circuit addresses the data and/or instructions in the main memory.” Ex. 1001, 8:2–7.

Petitioner contends that under either party's definition of the "data and/or instructions" limitations,⁹ Irie in view of AAPA teach the subject matter of claim element 7[c] and that a POSITA would have implemented Irie's cache memory in accordance with the known functionality of cache described in the AAPA. Pet. 30–39 (citing Ex. 1001, 1:34–40, 2:35–52; Ex. 1002 ¶¶ 9–14, 23–32, Fig. 1; Ex. 1008 ¶¶ 46, 73–92; Ex. 1019, 111, Fig. 3.2; Ex. 1024, 2:54–67). Petitioner asserts Irie teaches using cache memory (cache memory 11) and main memory (including ROM 21) in a normal operating mode. Pet. 30. Petitioner also asserts that AAPA describes one conventional implementation of cache memory functionality in which known cache memory and main memory are used "to cache in cache memory a part of data and/or instructions that the processing circuit addresses in the main memory during execution," as recited in claim element 7[c]. *Id.* at 30–31 (citing Ex. 1008 ¶ 76). In particular, Petitioner

⁹ For example, Petitioner contends the combination of Irie and AAPA teaches the "data and/or instructions" limitations under Petitioner's proposed construction because a POSITA would have understood that when the content of program 200 (data and/or instructions) are copied from ROM 21 to cache memory 11, Irie's CPU 10 addresses program 200 in ROM 21 (main memory address) corresponding to each instruction of program 200 on address bus 12 (address/data interface of CPU) so that the address can be provided to ROM 21). Pet. 33–34. Petitioner also asserts that Irie and AAPA teach the "data and/or instructions" limitations under Patent Owner's proposed construction, because a POSITA would have understood that Irie's CPU 10 identifies each portion of instructions of program 200 ("data and/or instructions") using its respective address in ROM 21 because each address of ROM 21 corresponds to each portion of instructions of program 200 and is used to copy the instructions stored at each address into cache memory 11 and that each ROM 21 address can be used to retrieve its respective instructions from ROM 21). *Id.* at 34–35. Patent Owner does not dispute these assertions.

contends that AAPA explains that known apparatuses have caches that store copies of part of data and/or instructions that a processor addressed in main memory to speed up execution. *Id.* at 30 (quoting Ex. 1001, 1:34–37, 2:35–52). Petitioner relies on the statement in the ’331 patent, that in these “known apparatuses,” the “cache memory temporarily stores copies of part of the data and/or instructions that the processor has addressed in main memory, so that it can be retrieved given its main memory address.” *Id.* at 30–31 (quoting Ex. 1001, 1:37–40). Petitioner also asserts that AAPA’s teaching that when “the processor addresses such data and/or instructions again, the cache memory substitutes the cached data and/or instructions for the data and/or instructions from main memory” satisfies the limitations requiring “substitut[ing] cached data and/or instructions when the instruction processing circuit addresses the data and/or instructions in the main memory.” *Id.* at 31 (citing Ex. 1001, 1:37–43; Ex. 1008 ¶ 77).

Petitioner contends a POSITA would have combined the teachings of Irie and AAPA because doing so results in combining known prior art elements according to known methods to yield predictable results and the combination uses a known technique to improve similar devices in the same way. Pet. 35–41. For example, Petitioner asserts that Irie’s CPU 10, cache 11, and main memory (ROM 21 and RAM 22) interact in a normal operating mode when the CPU addresses instruction content in main memory. *Id.* at 36. Petitioner further asserts that AAPA’s cache memory also interacts with the processor and main memory. *Id.* (citing Ex. 1001, 1:34–44). Petitioner contends that implementing Irie’s cache memory like AAPA’s cache memory, in Irie’s normal operating mode, would have simply entailed incorporating AAPA’s known cache functionality into Irie’s cache memory using known, conventional methods of memory design and

achieve predictable results with an expectation of success. *Id.* at 36–38 (citing Ex. 1001, 1:34–44, 2:35–52; Ex. 1002 ¶¶ 9–13; Ex. 1008 ¶¶ 88–89); *see also id.* at 37 (stating Irie’s cache memory 11 would still function the same way in the combination as it does separately) (citing Ex. 1002 ¶¶ 9–13; Ex. 1008 ¶ 88).

Petitioner further asserts a POSITA would have applied AAPA’s cache memory functionality to Irie’s cache memory to provide the known, predictable increase in execution speed. Pet. 38–39; Ex. 1001, 1:34–37 (stating known apparatus are often provided with a cache memory in order to speed up execution); Ex. 1002 ¶ 23 (describing operation of cache memory in normal operating mode “in order to improve the processing speed of . . . CPU 10”); *see also* Pet. 39 (stating a POSITA “would have modified Irie’s cache memory to operate like AAPA’s cache memory to provide a beneficial, predictable, increase in execution speed by reducing the delay of the availability of program instructions and data to Irie’s CPU 10”) (citing, *inter alia*, Ex. 1002 ¶¶ 11–14, 23–32; Ex. 1008 ¶ 91).

Patent Owner does not dispute in its Patent Owner Response or Sur-reply that the combination of Irie and AAPA teaches or suggests the subject matter recited in claim element 7[c]. *See generally* PO Resp. and PO Sur-reply (arguing that for this asserted ground, the cited art does not teach or suggest certain elements of claim 7, but making no specific argument regarding claim element 7[c]). Rather, Patent Owner argues that Petitioner improperly relies on AAPA “*in the* combination as a prior art reference directly combined with other references, just as the Guidance memo forbids.” Paper 41, 1.

Based on a review of the complete record, we determine Petitioner does not improperly rely upon AAPA in its unpatentability argument and

that Irie, in view of the knowledge of a POSITA as set forth in AAPA, teaches or suggests the subject matter of claim element 7[c].

Section 311(b) of Title 35 of the U.S. Code limits the prior art that may be used as the “basis” of an IPR proceeding to “patents or printed publications.” 35 U.S.C. § 311(b); Guidance Memo, 3. A patent cannot be prior art to itself, and thus, a patent challenged in an IPR cannot form the “basis” of an IPR. Guidance Memo, 3–4. For this reason, admissions by an applicant in the specification of a challenged patent, standing alone, cannot be used as the basis for instituting an IPR. *Id.* at 4 (citing 35 U.S.C. § 311(b)).

Notwithstanding the provisions of § 311(b), a properly conducted obviousness analysis necessarily depends upon the knowledge possessed by an ordinary-skilled artisan. *Id.* at 5. Statements in a challenged patent’s specification “can be used in an IPR to the extent they provide evidence of the general knowledge of those with ordinary skill in the art.” *Id.* The use of such statements in a specification “is consistent with § 311(b) because such statements can supply legally relevant information, while not constituting the ‘basis’ of the obviousness ground raised under § 103.” *Id.* at 6. Permissible uses of general knowledge of one having ordinary skill under § 103 include, *inter alia*, “supplying missing claim limitations that were generally known in the art prior to the invention” or “demonstrating the knowledge of the ordinary skilled artisan at the time of the invention.” *Id.* (citing *Koninklijke Philips N.V. v. Google LLC*, 948 F.3d 1330, 1337–38 (Fed. Cir. 2020)).

We determine Petitioner relies on statements in the ’331 patent (AAPA) to demonstrate the known, conventional behavior of cache memory, which is a permissible use of AAPA. *See* Pet. 17–19, 30–36; *see also*

Paper 39, 2–5. For example, Petitioner points to statements in the '331 patent that “[k]nown apparatuses that contain a computer processor and a main memory with data and/or instructions for use by the processor are often provided with a cache memory in order to speed up execution.” Pet. 17–18, 30 (quoting Ex. 1001, 1:34–47). Petitioner also points to the statement in the '331 patent that, in these known systems “[t]he cache memory temporarily stores copies of part of the data and/or instructions that the processor has addressed in main memory so that it can be retrieved given its main memory address” and “[w]hen the processor addresses such data and/or instructions again, the cache memory substitutes the cached data and/or instructions for the data and/or instructions from main memory.” *Id.* at 18, 30–31 (quoting Ex. 1001, 1:37–43). Petitioner contends that because AAPA states that “known caches store copies of part of data and/or instructions that a processor addressed in main memory to speed up execution,” AAPA discloses using known cache memory and main memory “to cache in cache memory a part of data and/or instructions that the instruction processing circuit addresses in the main memory during execution as claimed.” *Id.* at 31 (citing Ex. 1001, 1:34–40; Ex. 1008 ¶ 76).

Additionally, we disagree with Patent Owner’s contention that Petitioner uses AAPA as combinable prior art, not as evidence that element 7[c] was generally known in the art. Paper 41, 1, 11–15; *see also id.* at 11–12 (stating the Petitioner argues the function of the cache in limitation 7[c] “was ‘known’ in a prior art reference—the AAPA—and uses AAPA as prior art combinable with other prior art, not as alleged proof of ‘general knowledge’”). Petitioner’s statements that AAPA is prior art or that AAPA in combination with Irie teaches the subject matter of claim element 7[c], however, do not negate the fact that Petitioner relies on AAPA to evidence

the general knowledge possessed by someone of ordinary skill in the art.
See Guidance Memo, 4.

Patent Owner also contends that Petitioner never argues that the AAPA was within the general knowledge of a POSITA. Paper 41, 12–14; *see also id.* at 12 (stating Petitioner “did not even attempt to show that AAPA’s caching functionality in Irie’s alleged main memory normal operating mode . . . would have been ‘generally known’”); *id.* at 13 (stating the “Petition never showed that the AAPA or element 7[c] itself were within the general knowledge of the POSITA. Instead, the Petitioner relied on the AAPA to disclose that functionality”). Patent Owner contends that Petitioner merely states that the AAPA is “‘known’—in other words, in a single prior art reference” and does not state that the AAPA is “well-known.” *Id.* at 12–14 (citing Pet. 17, 31, 35–36, 38).

This argument is not persuasive as Petitioner relies on, *inter alia*, Dr. Sechen’s testimony that a “conventional and well known implementation of cache memory functionality is described by AAPA.” Ex. 1008 ¶ 76; *see also* Pet. 31 (citing Ex. 1008 ¶ 76); Pet. 30 (stating “AAPA describes one conventional implementation of cache memory functionality”); Pet. 18 (stating “AAPA recognizes the conventional behavior of its known apparatuses”). By asserting that AAPA describes a “conventional and well known implementation of cache memory functionality,” Petitioner has asserted that the cache functionality described in AAPA was generally known.

Thus, we determine Petitioner relies upon statements in the ’331 patent to evidence the general knowledge possessed by someone of ordinary skill in the art and that it is permissible for Petitioner to use the statements to support Petitioner’s obviousness argument. We also agree

with Petitioner’s contention that a POSITA would have combined the teachings of Irie with the general knowledge of the art as stated in AAPA because doing so would have simply entailed incorporating well-known cache functionality into Irie’s cache memory to achieve predictable results (e.g., Irie would have addressed information in ROM 21 using address bus 13 and Irie’s cache memory 11 would have operated to cache and substitute information as the conventional cache described in AAPA does in normal operation) with a reasonable expectation of success. *See* Pet. 35–38; Ex. 1008 ¶¶ 86–89.

Thus, based on the complete record and for the foregoing reasons, we determine Petitioner has sufficiently shown Irie and the general knowledge of a person of ordinary skill in the art, as evidenced by AAPA, teach or suggest the claimed subject matter of claim element 7[c] and has articulated sufficient reasoning with rational underpinning for modifying Irie to arrive at the subject matter of claim element 7[c].

d) Element 7[d]

Claim element 7[d] recites: “storing, in the main memory, a program of instructions for executing an interrupt function during operating in a low power operating mode.” Ex. 1001, 8:8–10.

Petitioner asserts Irie teaches the subject matter recited in claim element 7[d]. Pet. 39–41 (citing Ex. 1002 ¶¶ 24–25, 27–32, 36–37, Figs. 2–4; Ex. 1008 ¶¶ 94–96). For example, Petitioner asserts Irie teaches storing an “interrupt processing program” in main memory (e.g., ROM 21). *Id.* at 40 (citing Ex. 1002 ¶ 27).

Petitioner also asserts Irie’s “interrupt processing program,” which is executed when CPU 10 is in an energy-savings mode, is “a program of instructions for executing an interrupt function during operating in a low

power operating mode” as recited in claim element 7[d]. *Id.* at 39–40 (citing Ex. 1002 ¶¶ 25–37; Ex. 1008 ¶ 94). In particular, Petitioner asserts Irie teaches that CPU 10 is placed in a low power operating mode when power switch 34 is turned to “off,” which causes signal 34 to move “from H level to L level,” which, in turn, places CPU 10 “in an ‘energy-saving mode’ where ‘power to peripheral devices of . . . CPU 10 [including] circuits 24 and 25 and . . . memories 21 and 22 . . . is turned off.’” *Id.* at 39–40 (citing Ex. 1002 ¶¶ 24–36). Petitioner contends that when the CPU is in this “energy-savings mode” and power switch 34 is turned back to “on,” signal S34 moves from L level to H level. *Id.* at 40 (citing Ex. 1002 ¶ 37). Petitioner contends that when the change in signal S34 happens, “CPU 10 is interrupted, and [Irie’s] interrupt program” 200 is executed. *Id.* (citing Ex. 1002 ¶ 37); *see also* Ex. 2012, 25:5–11 (Dr. Sechen stating that in Irie “pressing a keyboard key or something would cause . . . switch movement [that] is detected by the CPU, which would then essentially cause it to execute the program that’s stored in the cache, and that program would bring the CPU back to full life, that is, full power and normal mode”).

Patent Owner responds that (1) Irie’s “interrupt processing program” is merely a “wake up” program, not an “interrupt function” and (2) Irie’s interrupt processing program is not executed “during operating in a low power operating mode” as required by claim element 7[d]. PO Resp. 4–16; PO Sur-reply 1–7.

Based upon our review of the complete record, we agree with Petitioner’s persuasive argument and evidence that Irie’s interrupt processing program is stored in main memory (ROM 21) and is a program of instructions for executing an interrupt function during operating in a low power operating mode, notwithstanding Patent Owner’s arguments to the

contrary. *See, e.g.*, Pet. 39–41 (citing Ex. 1002 ¶¶ 24–25, 27–32, 36–37, Figs. 2–4; Ex. 1008 ¶¶ 94–96).

For example, we agree with Petitioner’s argument that Irie’s “interrupt processing program” 200 is a program of instructions for executing an interrupt function that is executed when CPU 10 is in an “energy-savings mode” as a result of power switch 34 turning “on.” Pet. 39–40 (citing Ex. 1002 ¶¶ 27, 36, 37; Ex. 1008 ¶ 94). In particular, paragraph 37 of Irie states:

When the power switch 34 is turned from off to on, the signal S34 goes from L level to H level. When this happens, *the CPU 10 is interrupted, and the interrupt program copied to the cache area 11A is executed in step 106. As a result*, the process moves from step 204 to step 211, and the CPU 10 returns to normal mode.

Ex. 1002 ¶ 37 (emphases added); *see also* Ex. 2012, 25:5–11 (Dr. Sechen stating that in Irie pressing a keyboard key causes switch movement that is detected by Irie’s CPU, which causes execution of the interrupt processing program that brings the CPU back to normal mode).

We disagree with Patent Owner’s contention that Irie’s process of waking up the CPU by pressing a switch is not an “*interrupt function*” as required by claim 7. *See* PO Resp. 4–15; PO Sur-reply 1–5. Patent Owner contends that Irie’s waking up a computer to return to normal operating mode “is significantly different from the claimed invention’s goal of enabling the execution of interrupt functions in low power mode without a need to return to normal operating mode.” PO Resp. 7–8 (citing Ex. 1002 ¶¶ 9, 34, 37, 45); *see also id.* at 7–15 (asserting Irie’s interrupt program simply wakes up Irie’s terminal from low power operating mode so functions can be performed in normal operating mode and thus does not

teach executing an interrupt function in low power operating mode); *id.* at 11 (stating “simply waking up the system from a low power mode is not the ‘function’ claimed and described” in the ’331 patent); *see also id.* at 12 n.2 (stating the ’331 patent “focus[es] on allowing, to the extent possible, performance of functions during the low power mode” and that this “feature is fundamentally different from Irie, which always must return the system to normal operating mode to perform any functions, and whose ‘wake up’ program exists *solely* to return the system to normal operating mode so operation can resume” (emphasis added)).

Contrary to Patent Owner’s assertions, the ’331 patent expressly describes embodiments in which an interrupt program returns the system to its normal operating mode, including by the activation of a switch. For example, the ’331 patent expressly states that “examples of [an interrupt] function include processing interrupts caused by user activation of control switches.” Ex. 1001, 1:18–20. The ’331 patent further explains that “in practice the apparatus will be in the low power mode for long periods of time” and that is “desirable that power consumption in this low power mode is minimized.” *Id.* at 1:26–28. The ’331 patent goes on to state that minimizing power consumption is important for apparatuses, “such as mains operated television sets that are switched to standby in view of the extended period of times that such apparatuses remain in the standby mode.” *Id.* at 1:28–33.

Additionally, the ’331 patent expressly states that “actuation of a user interface button” can trigger the processor to execute the interrupt program (Ex. 1001, 3:55–60) and that if “execution of the interrupt program results in a need to return to the normal operating mode, processor 14 will execute [a step] of the interrupt program to cause main power supply circuit 10 to

reactivate” (Ex. 1001, 3:67–4:14, Fig. 2 (steps 26, 27, 28)). In particular, the ’331 patent explains that in the first step (step 26) of its interrupt program, process 14 addresses the instructions of the interrupt program that are stored in cache memory, and is therefore able to execute the interrupt program without recourse to main memory. *Id.* at 3:64–67. If execution of the interrupt program results in the need to return to the operating mode (step 27), the processor will execute a third step (step 28) of the interrupt program *that causes the main power supply to reactivate*. *Id.* at 3:67–4:5. If, however, it is not necessary to return to the normal operating mode, “processor 14 will execute further steps 29, *if any*, of the interrupt program and main memory will remain deactivated during the entire execution of the interrupt program. *Id.* at 3:67–4:9 (emphasis added). For example, the apparatus can remain in the low power operating mode, if the interrupt program determines that no valid button was actuated. *Id.* at 4:10–13. Thus, the ’331 patent states if the interrupt program results in the execution of steps 26, 27, and 28 (and operating mode step 29 is not executed), main power supply circuit 10 reactivates in step 28 and there is a return to normal operating mode without the continued execution of interrupt program in low power mode as set forth in step 29. As such, the ’331 patent expressly describes executing an interrupt program that returns the system to its normal operating mode.

We also disagree with Patent Owner that Irie’s program of instructions for executing an interrupt function is not executed *in a low power operating mode*. PO Resp. 10, 15–16; PO Sur-reply 1–5. Patent Owner contends that Irie’s interrupt processing program 200 is only *partially* executed during operating in low power mode, because the interrupt program returns the system to the normal mode in the initial step

(i.e., step 211) of the program, and therefore Irie's interrupt program is not executed "during operating in" the low power operating mode. PO Resp. 15 (citing Ex. 1002 ¶¶ 27–39, Fig. 3; Ex. 2012, 22:24–23:6; Ex. 2013 ¶ 43). Patent Owner again contends that a main goal of the '331 patent is that the apparatus can perform certain basic functions in the lower power mode, without switching to the normal operating mode. *Id.* at 15–16 (citing Ex. 1001, 1:16–25, 3:35–61, 4:5–9).

We disagree with this argument for two reasons. First, claim 7 does not exclude an interrupt function that returns the system to a normal operating mode. As noted above, the Specification of the '331 patent describes embodiments in which execution of the interrupt program results in reactivating the system and a return to normal mode. *See, e.g.*, Ex. 1001, 3:64–4:16, Fig. 2 (steps 26, 27, and 28). *See Network-1 Techs., Inc. v. Hewlett-Packard Co.*, 981 F.3d 1015, 1024 (Fed. Cir. 2020) ("A claim interpretation that excludes a preferred embodiment from the scope of the claim is rarely, if ever, correct." (quoting *MBO Labs., Inc. v. Becton, Dickinson & Co.*, 474 F.3d 1323, 1333 (Fed. Cir. 2007))). "[W]here claims can reasonably [be] interpreted to include a specific embodiment, it is incorrect to construe the claims to exclude that embodiment, absent probative evidence on the contrary." *GE Lighting Sols., LLC v. AgiLight, Inc.*, 750 F.3d 1304, 1311 (Fed. Cir. 2014) (quoting *Oatey Co. v. IPS Corp.*, 514 F.3d 1271, 1277 (Fed. Cir. 2008)). Here, Patent Owner does not point to, nor do we discern, any compelling evidence to the contrary, such as statements made during prosecution or in the Specification that indicate patentee's intent to exclude from the scope of claim 7 an interrupt function that returns the system to a normal operating mode.

Second, we disagree with Patent Owner that Irie's interrupt processing program 200 is only partially executed during operating in low power operating mode because Irie enters normal mode in step 211 of program 200. PO Resp. 15–16 (citing Ex. 1002 ¶¶ 37–39, Fig. 3). Rather, we agree with Petitioner that Irie's interrupt processing program is executed when the CPU is in an energy-savings mode (the claimed “low power operating mode”) and Irie does not return to a normal operating mode until step 115, which occurs *after* execution of interrupt processing program 200. *See, e.g.*, Pet. 39–41 (citing, *inter alia*, Ex. 1002 ¶¶ 24–37, Figs. 2–4); Ex. 1002 ¶ 41; Tr. 22:18–22, 88:1–6. Petitioner contends that Irie enters an “energy-savings mode” where power to peripheral devices of CPU 10, including circuits 24 and 25, is turned off. Pet. 39–40 (citing Ex. 1002 ¶¶ 24–36); *see also* Ex. 1002 ¶ 36 (stating that when CPU 10 enters standby mode, power to peripheral devices, including circuits 24 and 25, is turned off and “[a]ccordingly, in this state, the current consumed when . . . CPU 10 is [in] an energy-saving mode can be saved”). Irie also explains that “in step 115, peripheral devices, such as . . . circuits 24 and 25, etc., are initialized, after which *normal mode is returned to*, and operation begins.” Ex. 1002 ¶ 41 (emphasis added). Thus, we agree with Petitioner that Irie's CPU is in an energy-saving mode (the claimed “low power operating mode”) when power to peripheral devices, such as circuits 24 and 25, is turned off. Power to peripheral devices 24 and 25 is not restored until step 115, which occurs after execution of interrupt processing program 200.

As such, we agree with Petitioner's persuasive argument and evidence that Irie's interrupt processing program 200, which is executed before step 115 during which Irie returns to normal power mode, is a “program of instructions for executing an interrupt function during operating in a low

power mode” and that Petitioner has sufficiently shown that Irie teaches the subject matter recited in claim element 7[d]. *See, e.g.*, Pet. 39–40; Ex. 1002 ¶¶ 24–37, 41; Ex. 1008 ¶ 94).

e) Element 7[e]

Petitioner asserts the storing of Irie’s interrupt processing program at the “starting address, final address, and intervening addresses” in ROM 21 satisfies the limitations of claim element 7[e], which recites “wherein the interrupt program is stored at addresses in main memory that have been selected so that all instructions of the interrupt program can be stored together in the cache memory.” Pet. 41–44 (citing Ex. 1002 ¶¶ 27–37, Figs. 2–4; Ex. 1008 ¶¶ 97–101); Ex. 1001, 8:11–14. Dr. Sechen, Petitioner’s declarant, explains that Irie starts the process of copying the interrupt processing program from ROM 21, via routine 300, at the “first starting address,” which is the first address where the instructions of the interrupt processing program have been stored in ROM 21. *See* Ex. 1008 ¶ 101 (citing Ex. 1002 ¶ 28). Routine 300 then copies the instruction content at that address to a temporary location in RAM 22 and then copies the content from the temporary location to area 11A of cache memory 11. *Id.* (citing Ex. 1002 ¶¶ 24–32, Figs. 2–4). The first starting address of ROM 21 is next “incremented by 1” thereby identifying for copying the next address at which instruction content was stored, and the same process of copying instruction content is repeated until content from a “final address” of ROM 21 is copied to area 11A of cache memory 11. *See id.* (citing Ex. 1002 ¶¶ 28–30). Dr. Sechen testifies that a POSITA would have understood that this process describes “wherein the interrupt program is stored at addresses in main memory that have been selected so that” all instructions of Irie’s

interrupt processing program are stored together in cache memory 11 or a single part of cache memory (cache area 11A) at the same time. *Id.*

Patent Owner responds that Irie does not teach storing addresses in main memory “that have been selected so that all instructions of the interrupt program can be stored together in cache memory” because Irie does not teach storing the addresses *contiguously*. PO Resp. 1, 17– 39.

We disagree with Patent Owner because its argument is based upon a claim construction that we do not adopt. As stated above in § II.B.1, we construe the phrase “wherein all instructions of the interrupt program are stored together in the cache memory” as encompassing storage of all instructions of an interrupt program *in part of* a cache memory, such as the part of cache memory that is not deactivated when other parts of cache memory remain activated. Petitioner has persuasively shown that Irie teaches that when routine 300 is executed, program 200, which is in ROM 21, is sequentially copied to the part of cache memory designated *cache area 11A* and not to the part of cache memory designated *cache area 11B*. *See, e.g.*, Pet. 41–44 (citing Ex. 1002 ¶¶ 27–37; Ex. 1008 ¶¶ 97–101). For example, Petitioner provides persuasive evidence and argument, supported by the declaration testimony of Dr. Sechen, that the starting, final, and intervening addresses in ROM 21 are selected so that the interrupt program could be copied to and stored in cache memory 11A at the same time using copying routine 300. *Id.* at 42–43 (citing Ex. 1008 ¶ 101; Ex. 1002 ¶¶ 28–32, Figs. 2–4). Thus, Petitioner has sufficiently shown that Irie teaches the subject matter recited in claim element 7[e].

f) Element 7[f]

Petitioner asserts Irie teaches “detecting that it is no longer necessary to operate in the normal operating mode” as recited by claim element 7[f].

Pet. 44–48 (citing Ex. 1002 ¶¶ 18–19, 24–37; Ex. 1008 ¶¶ 102–108); Ex. 1001, 8:15–16. For example, Petitioner contends that when Irie’s power switch 34 is turned off, CPU 10 recognizes that signal S34 changes “from H level to L level,” which causes CPU 10 to execute routine 100, which then causes CPU 10 to switch from normal operating mode to an energy saving mode. *See id.* at 44–48 (citing Ex. 1002 ¶¶ 18–19, 24–37; Ex. 1008 ¶¶ 102–108).

We agree with Petitioner’s analysis and find that, for the reasons explained by Petitioner, Irie teaches the subject matter recited in claim element 7[f]. *See* Pet. 44–48; Ex. 1008 ¶¶ 102–108; Ex. 1002 ¶¶ 18–19, 24–37. Patent Owner has not raised arguments against this limitation in its Patent Owner Response; therefore any such arguments are waived.

g) Element 7[g]

Claim element 7[g] recites “switching to the low power operating mode once it is detected that it is no longer necessary to operate in the normal operating mode, by loading the interrupt program into the cache memory from the main memory, wherein all instructions of the interrupt program are stored together in the cache memory.” Ex. 1001, 8:17–24.

Petitioner asserts Irie’s disclosure of switching from a normal operating mode to an energy saving mode after CPU 10 recognizes signal S34 changing from “H level to L level” teaches “switching to the low power operating mode once it is detected that it is no longer necessary to operate in the normal operating mode.” Pet. 48–49 (citing Ex. 1002 ¶¶ 24–36, 42; Ex. 1008 ¶¶ 109–110). Petitioner also contends Irie discloses “loading the interrupt program into cache memory from the main memory” by teaching that the “interrupt program” is copied from ROM 21 to RAM 22 and then from RAM 22 to cache 11. *Id.* at 49–50 (citing Ex. 1002 ¶¶ 24–36;

Ex. 1008 ¶ 111). Specifically, Petitioner asserts that in step 106 of routine 100 “programs and tables which are needed for later processes are read into memory 11.” *Id.* at 49 (citing Ex. 1002 ¶¶ 27–28). Petitioner further contends that when routine 300 is executed, the CPU copies program content stored at a “starting address location in ROM 21” to a temporary location in RAM 22; the instruction content is then moved from the temporary location in RAM 22 to cache memory 11. *Id.* (citing Ex. 1002 ¶¶ 27–31). The CPU determines a “subsequent address location in ROM 21” where further program instruction content is stored, copies this content to the same temporary location in RAM 22, and then these instructions are moved from the temporary location in RAM 22 to cache memory 11. *Id.* (citing Ex. 1002 ¶¶ 28–29). This process is continued until a final address holding instruction content in ROM 21 is reached. *Id.* (citing Ex. 1002 ¶¶ 29–30).

Petitioner also asserts that Irie teaches that “all instructions of the interrupt program are stored together in the cache memory,” as recited in claim element 7[g], for the same reason Irie teaches “so that all instructions of the interrupt program can be stored together in cache memory” as recited in claim element 7[e]. *See* Pet. 50 (citing Ex. 1008 ¶ 112).

Patent Owner responds that Irie does not teach “all instructions of the interrupt program are stored together in the cache memory” recited in claim element 7[g] for the same reasons Irie does not teach “so that all instructions of the interrupt program can be stored together in cache memory” as recited in claim element 7[e]. *See, e.g.*, PO Resp. 1, 30–31. Patent Owner’s arguments are not persuasive for the same reasons set forth above in connection with claim element 7[e]. We have reviewed and agree with Petitioner’s argument and evidence supported by Dr. Sechen’s declaration

testimony, and described in detail above, that Irie teaches the subject matter recited in claim element 7[g]. Other than Patent Owner's argument that Irie does not teach the limitations of claim element 7[g] for the same reasons set forth with respect to claim element 7[e], Patent Owner does not dispute that Irie teaches the remaining subject matter recited in claim element 7[g], and has therefore waived any such argument.

h) Element 7[h]

Claim element 7[h] recites "switching to the low power operating mode once it is detected that it is no longer necessary to operate in the normal operating mode, by . . . deactivating the main memory to reduce power consumption, but keeping active at least a part of the cache memory, that is needed for retrieving the interrupt program and for executing the interrupt function." Ex. 1001, 8:17–19, 7:25–28. Petitioner asserts Irie's disclosure of turning off power to peripheral devices (e.g., ROM 21 and RAM 22) while keeping cache area 11A active when the mobile information terminal transitions from a normal operating mode to an energy-saving mode, discloses this limitation. Pet. 50–52 (citing Ex. 1002 ¶¶ 24–36). Petitioner contends cache area 11A is kept active so that the interrupt routine copied to cache area 11A can be read from cache memory when the terminal needs to exit the energy-saving mode. *Id.* at 52–53 (citing Ex. 1002 ¶¶ 36–37; Ex. 1008 ¶ 119); *see also id.* at 50–54 (citing Ex. 1008 ¶¶ 114–121).

We agree with Petitioner's analysis and determine that, for the reasons explained by Petitioner, Petitioner has shown sufficiently that Irie teaches the subject matter recited in claim element 7[h]. Patent Owner has not raised arguments against this limitation in its Patent Owner Response; therefore any such arguments are waived.

i) Element 7[i]

Petitioner contends Irie's disclosure of executing its interrupt program from cache area 11A teaches the limitations of claim element 7[i], which requires "executing the interrupt program from said at least part of the cache memory." *See* Pet. 54 (citing Ex. 1002 ¶¶ 32, 35–36, 37; Ex. 1008 ¶¶ 122–123); Ex. 1001, 8:29–30.

We agree with Petitioner's analysis and determine that, for the reasons explained by Petitioner, Petitioner has shown sufficiently that Irie teaches the subject matter recited in claim element 7[i]. Patent Owner has not raised arguments against this limitation in its Patent Owner Response; therefore any such arguments are waived.

j) Conclusion

For the foregoing reasons and based on the complete record, we determine Petitioner has sufficiently shown that Irie and AAPA teach or suggest all of the limitations of claim 7 and that a person of ordinary skill in the art would have had reason to modify Irie in view of AAPA in the manner asserted. Thus, for the reasons discussed above, Petitioner has demonstrated by a preponderance of the evidence that claim 7 would have been obvious over Irie in view of AAPA.

E. Asserted Unpatentability of Claim 7 over Irie, AAPA, and Bourekas

Petitioner alternatively contends that, to the extent *stored together* in the cache memory is determined to mean *stored in contiguous locations* in the cache memory, claim 7 would have been obvious under 35 U.S.C. § 103(a) over Irie in view of AAPA and Bourekas. *See* Pet. 54–64; Pet. Reply 22–26. Patent Owner opposes. PO Resp. 39–54; PO Sur-reply 17–22.

1. *Overview of Bourekas*

Bourekas is titled “Direct-Mapped Cache with Cache Locking Allowing Expanded Contiguous Memory Storage by Swapping One or More Tag Bits with One or More Index Bits.” Ex. 1005, code (54). In a direct mapped cache, the cache is accessed by index (i.e., the index serves as an address where the data word and its tag are stored in the cache memory). *Id.* at 1:31–33. Thus, in a direct mapped cache, an address in main memory can be mapped to only one particular cache location. *Id.* at 1:38–40. In some applications, the tags of critical data words are stored in the cache and “locked” to prevent them from being deleted so that the data can be accessed as quickly as possible. *Id.* at 1:57–60. Bourekas states that it is difficult to use cache locking in a direct mapped cache for software greater than 2kB. *See id.* at 2:28–28. Bourekas states that previous solutions to cache locking large software include (1) adding complex page management software, (2) using a “translation lookaside buffer (“TLB”) that increases hardware complexity and performance degradation and additional operating system software, or (3) implementing cache locking in “set associative caches,” which are also more complex and costly than direct mapped caches. *See id.* at 2:28–55. Bourekas states that its method involves, *inter alia*, dividing the direct mapped cache memory into two equal halves, each half servicing a contiguous address range of main memory. *Id.* at 2:64–3:1. A programmer can store critical software in one of the contiguous portions of main memory and lock it into one half of the cache. *Id.* at 3:4–6. The programmer can then store other software in the other contiguous portion of the main memory, which is serviced by the other half of the cache. Thus, a direct mapped cache with cache locking is realized without a TLB or additional

page management or operating system software. *See id.* at 3:4–11; *see also id.* at 3:65–4:5, 4:46–64, 4:21–24, Fig. 3

2. *Analysis of Claim 7*

Petitioner asserts Irie and AAPA teach the limitations of 7[a]–[d], 7[f], 7[h], and 7[i] for the same the reasons asserted in Petitioner’s first challenge based on Irie and AAPA, discussed above. Pet. 54 (citing Ex. 1008 ¶ 124). Petitioner asserts that to the extent the “addresses in main memory that have been selected so that all instructions of the interrupt program can be stored *together* in the cache memory” is determined to require that the addresses in main memory are selected so that all instructions of the interrupt program can be stored together in *contiguous memory addresses* of a cache memory at the same time, then Irie in view of Bourekas teaches this narrower interpretation and satisfies the limitations of claim elements 7[e] and [g]. *See id.* at 55–64 (citing Ex. 1008 ¶¶ 125–150).

For example, Petitioner contends the direct mapping of Bourekas’ main memory to cache memory allows storage at contiguous cache locations; for example, Bourekas’ direct mapping technique allows a user to lock a “4 word program” stored at physical address range 0000 to 0011 of the main memory at contiguous cache locations having address indexes 000, 001, 010, and 011, respectively. *See id.* at 56 (citing Ex. 1003, 3:46–4:9, Fig. 3; Ex. 1008 ¶ 129). Petitioner asserts that a POSITA would have known that storing Irie’s interrupt program in cache with contiguous cache locations by using well-known direct cache mapping techniques, such as those taught by Bourekas, would allow for “less complexity” and improved control, known benefits when designing memory systems. *See id.* at 57–58, 62 (citing Ex. 1005, 2:25–3:20, 4:10–20; Ex. 1008 ¶¶ 129–131). Petitioner asserts Bourekas explains the benefits of direct mapping include a “direct

mapped cache without TLBs, additional sets of tag comparators, or additional page management or operating system software.” *Id.* at 58 (citing Ex. 1005, 4:10–20; Ex. 1019, 546, 570, 574–75; Ex. 1023, 397–99, Fig. 5.4; Ex. 1008 ¶¶ 132–135). Petitioner further contends a POSITA would have been motivated to combine Irie and AAPA with Bourekas’ known technique to yield the predictable result of achieving Irie’s stated goal of moving an entire interrupt processing program from main memory to a specific part of cache memory. *Id.* at 59–62; *see also id.* at 62–63 (stating a POSITA would have applied Bourekas’ known technique to Irie’s similar device to obtain the improved and predictable result of a cache with a predictable, advantageous configuration).

Patent Owner responds that Petitioner has not demonstrated a reason why a POSITA would store Irie’s interrupt processing wake up program in contiguous address spaces within its cache 11A. PO Resp. 39–54; *see also* PO Sur-reply 18 (stating Irie does not express a preference to store the interrupt program continuously in cache 11A and nothing in Irie suggests that it would perform better if the interrupt program is stored contiguously in cache 11A). Patent Owner contends that Bourekas is directed to a highly specific problem in a specific type of cache memory where information is “locked” in cache memory to prevent the information from being deleted. *See, e.g.*, PO Resp. 40–41 (citing, *inter alia*, Ex. 1005, 1:38–40, 1:56–57, 1:64–2:28). Patent Owner contends Irie does not require any “locking” of data in cache memory and Petitioner has not demonstrated any reason for any program to be locked in Irie’s cache in the low power state. *See, e.g., id.* at 42–44. Patent Owner further asserts the purported benefits of combining Irie with Bourekas have no application in Irie because Irie does not have the problem of non-contiguous cache locking (*id.* at 45–52; PO Sur-reply 18–

19), and Petitioner's expert confirms the absence of any motivation to combine (PO Resp. 52–54).

We begin by noting that this ground is based upon a construction of claim 7 that requires the instructions be stored *contiguously*, a claim construction that we did not adopt. *See* § II.B.1. However, if claim 7 were construed to require that the addresses in main memory are selected so that all instructions of the interrupt program can be stored together in *contiguous memory addresses* of a cache memory at the same time, we find Petitioner has sufficiently shown the combination of Irie, AAPA, and Bourekas teaches each of the limitations of claim 7 under such a construction and that a person of ordinary skill in the art would have had reason to modify the Irie in view of AAPA and Bourekas in the manner asserted.

For example, we agree with Petitioner's assertions, which Patent Owner does not dispute, that Irie teaches storing the interrupt processing program in ROM 21, which is a main memory. *See* Pet. 55–56 (citing Ex. 1002 ¶¶ 27, 28, 32). We also agree with Petitioner's assertion, which Patent Owner does not dispute, that Bourekas teaches a direct mapping technique that maps main memory and cache memory to each other resulting in storage “at contiguous cache locations.” *See id.* at 56 (citing Ex. 1005, 3:46–4:9, 4:21–24, Fig. 3; Ex. 1008 ¶¶ 129–131). Petitioner also persuasively argues that a POSITA would have known that one reasonable way to store information in cache is to use direct cache mapping with contiguous cache locations because it allows for less complexity and improved control. *Id.* at 57–58 (citing Ex. 1005, 2:25–3:20). We agree with Petitioner's contention that a POSITA would have combined Bourekas' direct mapping technique to accomplish Irie's stated goal of moving an entire interrupt program from main memory to a specific part of cache

memory (i.e., cache area 11A) because the combination uses prior art memory elements according to known methods to yield predictable results. *Id.* at 59–62 (citing Ex. 1005, 3:46–55, 3:65–4:20; Ex. 1002 ¶¶ 9–13, 24–32; Ex. 1008 ¶¶ 139–144; Ex. 1019, 546). Petitioner also provides persuasive evidence and argument, supported by the declaration testimony of Dr. Sechen, that a POSITA would have understood that the combination would have provided (1) a less complex system that would not require “TLBs, additional sets of tag comparators, or additional page management or operating system software” and (2) increased control because users would have had the ability to control how and where information is cached. *Id.* at 57–64 (citing Ex. 1005, 2:25–3:20, 4:10–20; Ex. 1008 ¶¶ 125–150; Ex. 1019, 546, 570, 574–75; Ex. 1023, 397–99, Fig. 5.4).

We disagree with Patent Owner’s argument that Petitioner fails to show a motivation to combine Irie and Bourekas to store instructions together (i.e., contiguously) in cache. PO Resp. 39–54; PO Sur-reply 17–24. Patent Owner contends that Bourekas is directed to cache “locking” problems that have no application to Irie because Irie does not involve cache locking. PO Resp. 40–44. We disagree. Petitioner has persuasively shown that Bourekas discloses a direct mapping scheme as well as a technique for cache locking and that a POSITA would have understood that direct mapping can be implemented in a system that does not use cache locking. *See, e.g.*, Ex. 1032 ¶¶ 63–65; Ex. 2012, 63:12–13, 80:2–81:2, 86:14–23. We agree with Petitioner that a POSTIA would have understood that Irie’s system would benefit from Bourekas’ direct mapping approach, regardless of whether cache locking was implemented. Pet. 55–63; Ex. 1032 ¶¶ 65–66.

We also disagree with Patent Owner’s contention that Petitioner’s expert confirms the absence of any motivation to combine. PO Resp. 52–53.

Dr. Sechen persuasively testified that a POSITA would understand that Bourekas' direct mapping solution would benefit a system that does not provide locking by, *inter alia*, eliminating the need to use a TLB on-the-fly when implementing Irie's disclosure of storing instructions only in cache 11A and by eliminating the need for tag comparators when using set-associative mapping. *See, e.g.*, Ex. 2012, 80:10–19, 86:14–23, 116:16–24; Ex. 1008 ¶¶ 137–149; Ex. 1032 ¶¶ 71–73, 132; Ex. 1005, 2:49–51 (stating “set associative caches are more complex and costly than direct mapped caches because each set requires sense amplifiers and tag comparators”). We disagree with Patent Owner's contention that Irie would not realize Bourekas' benefits because Irie does not use a “set associative cache.” PO Resp. 47–49. Petitioner persuasively explains that although Irie does not describe what particular type of cache it uses, there are limited types of cache and that a POSITA would have reason to implement Bourekas' direct mapping technique over the other cache mapping techniques. *See, e.g.*, Ex. 1032 ¶ 132.

Thus, for the foregoing reasons and based on the complete record, we determine that if claim 7 were construed to require that the addresses in main memory are selected so that all instructions of the interrupt program can be stored together in *contiguous memory addresses* of a cache memory at the same time, we find Petitioner has sufficiently shown that Irie, AAPA, and Bourekas teach or suggest all of the limitations of claim 7 and that a person of ordinary skill in the art would have had reason to modify the Irie in view of AAPA and Bourekas in the manner asserted. Thus, for the reasons discussed above, Petitioner has demonstrated by a preponderance of the evidence that claim 7 would have been obvious over Irie, AAPA, and Bourekas.

III. CONCLUSION¹⁰

For the reasons discussed above, we determine Petitioner has proven, by a preponderance of the evidence, that the challenged claim of the '331 patent is unpatentable. The chart below summarizes our conclusions regarding the challenged claim.

Claims	35 U.S.C.	References/Basis	Claims Shown Unpatentable	Claims Not Shown Unpatentable
7	§ 103(a)	Irie, AAPA	7	
7	§ 103(a)	Irie, AAPA, Bourekas	7	
Overall Outcome			7	

IV. ORDER

Accordingly, it is

ORDERED that claim 7 of U.S. Patent No. 7,523,331 B2 has been shown to be unpatentable; and

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

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

IPR2019-01192
Patent 7,523,331 B2

For PETITIONER:

John Hobgood
Richard Goldenberg
Dominic Massa
WILMER CUTLER PICKERING HALE & DORR LLP
john.hobgood@wilmerhale.com
richard.goldenberg@wilmerhale.com
dominic.massa@wilmerhale.com

For PATENT OWNER:

Kenneth Weatherwax
Bridget Smith
Flavio Rose
Edward Hsieh
Parham Hendifar
Patrick Maloney
Jason Linger
LOWENSTEIN AND WEATHERWAX LLP
weatherwax@lowensteinweatherwax.com
smith@lowensteinweatherwax.com
rose@lowensteinweatherwax.com
hsieh@lowensteinweatherwax.com
hendifar@lowensteinweatherwax.com
maloney@lowensteinweatherwax.com
linger@lowensteinweatherwax.com