

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

ADVANCED MICRO DEVICES, INC.
Petitioner

v.

COLLABO INNOVATIONS, INC.
Patent Owner

Case IPR2019-01523
Patent 7,930,575 B2

PETITION FOR *INTER PARTES* REVIEW

Mail Stop "PATENT BOARD"
Patent Trial and Appeal Board
U.S. Patent & Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	MANDATORY NOTICES	2
	A. Real parties-in-interest	2
	B. Notice of related matters	3
	C. Lead and back-up counsel with service information	3
III.	GROUND FOR STANDING.....	3
IV.	STATEMENT OF RELIEF REQUESTED	4
V.	IDENTIFICATION OF CHALLENGE	4
VI.	TECHNICAL BACKGROUND	6
	A. Power management has been a longstanding concern in the computer industry.....	7
	B. In the mid '90s, a power-management standard, known as ACPI, was released.....	8
	C. The prior art discloses logic circuitry for preventing unnecessary current from flowing and for preventing the destruction of information during a shutdown event	9
VII.	THE '575 PATENT	11
	A. Overview	11
	B. Summary of the Prosecution History	11
	C. Level of Ordinary Skill in the Art	12
	D. Claim Construction.....	13
	1. "Microcontroller" (claims 1, 17).....	14
	2. "CPU" (claims 1, 17)	15
	3. "Power supply unit" (claims 1, 17).....	16

Inter Partes Review of U.S. Patent No. 7,930,575 B2

4.	“Power supply device” (claims 1, 17)	17
5.	“Power supply control unit” (claims 1, 17)	19
6.	“Information holding unit” (claims 1, 17)	20
7.	“Restores the information necessary in proceeding with the program and evacuated in the information holding unit in time of power shutdown” (claim 1)	21
8.	“The power supply control unit limits a period in which shutdown is possible” (claims 3, 19)	22
9.	“The plurality of microprograms”/ “a program for resetting processes, and the CPU selects one of the plurality of microprograms in an alternative way” (Claims 7, 22).....	23
10.	“A circuit for loading an inverted value to a register in a circuit portion in a power shutdown range” (claims 12, 27)	24
VIII. GROUNDS OF REJECTION		24
A.	<u>Ground 1</u> : Claims 1-3, 5-7, 11, 15, and 16 are obvious over Jahagirdar and ACPI	24
1.	Motivation to combine: Jahagirdar suggests the proposed combination.....	25
2.	Claim 1: The combination of Jahagirdar and ACPI renders this claim obvious.	27
3.	Claim 2: The combination of Jahagirdar and ACPI renders this claim obvious.	52
4.	Claim 3: The combination of Jahagirdar and ACPI renders this claim obvious.	53
5.	Claim 5: The combination of Jahagirdar and ACPI renders this claim obvious.	54

Inter Partes Review of U.S. Patent No. 7,930,575 B2

6.	Claim 6: The combination of Jahagirdar and ACPI renders this claim obvious.	55
7.	Claim 7: The combination of Jahagirdar and ACPI renders this claim obvious.	56
8.	Claim 11: The combination of Jahagirdar and ACPI renders this claim obvious.	61
9.	Claim 15: The combination of Jahagirdar and ACPI renders this claim obvious.	61
10.	Claim 16: The combination of Jahagirdar and ACPI renders this claim obvious.	62
B.	<u>Ground 2:</u> Claims 4, 17-22, 26, 30, and 31 are obvious over Jahagirdar, ACPI, and Saito.	63
1.	Motivation to combine Jahagirdar, ACPI, and Saito.	64
2.	Claim 4: The combination of Jahagirdar, ACPI and Saito renders this claim obvious.	67
3.	Claim 17: The combination of Jahagirdar, ACPI and Saito renders this claim obvious.	68
4.	Claims 18-22, 26, 30, and 31 are obvious in view of Jahagirdar, ACPI, and Saito.	70
C.	<u>Ground 3:</u> Claim 12 is obvious over Jahagirdar, ACPI, and Chung.	71
1.	Motivation to Combine Jahagirdar, ACPI, and Chung.	71
2.	Claim 12.	73
D.	<u>Ground 4:</u> Claim 27 is obvious over Jahagirdar, ACPI, Saito and Chung.	75

Inter Partes Review of U.S. Patent No. 7,930,575 B2

IX. CONCLUSION.....	76
APPENDIX A	III

Inter Partes Review of U.S. Patent No. 7,930,575 B2

PETITIONER’S EXHIBIT LIST

Exhibit No.	Description
1001	U.S. Patent No. 7,930,575 to Suginaka et al. (“’575 patent”)
1002	Prosecution History of U.S. Patent No. 7,930,575 (“’575 Prosecution History”)
1003	Declaration of Dr. Albonesi in Support of Petition for <i>Inter Partes</i> Review of U.S. Patent No. 7,930,575 (“Albonesi Decl.” or “EX1003”)
1004	Dr. Albonesi’s Curriculum Vitae
1005	U.S. Patent Pub. No. 2007/0157036 to Jahagirdar et al. (“Jahagirdar”)
1006	Compaq Computer Corporation <i>et al.</i> , “Advanced Configuration and Power Interface Specification, Revision 2.0” (July 27, 2000) (“ACPI”)
1007	U.S. Patent No. 5,390,350 to Chung <i>et al.</i> (“Chung”)
1008	U.S. Patent No. 5,530,879 to Crump <i>et al.</i> (“Crump”)
1009	Japanese Patent Appl. No.: S58-169218 to Saito.
1010	Certified English Translation of Japanese Patent Appl. No.: S58-169218 to Saito (“Saito”)
1011	U.S. Patent No. 6,141,762 to Nicol et al. (“Nicol”)
1012	Benini et al., “A Survey of Design Techniques for System-Level Dynamic Power Management”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 8, no. 3 (June 2000) (“Benini”)
1013	Grover, “Modern System Power Management,” Queue – Power Management, vol. 1, issue 7 (October 2003) (“Grover”)

Inter Partes Review of U.S. Patent No. 7,930,575 B2

Exhibit No.	Description
1014	Miller, M., “Windows 98 Put to the Test”, August 1, 1998 (“PC Magazine Report”) (also available at http://www.pcmag.com/article2/0,2817,1159610,00.asp)
1015	Collabo Innovation, Inc.’s Preliminary Infringement Contentions, <i>Collabo Innovations Inc., v. Advanced Micro Devices, Inc.</i> , 1:18-cv-552-LY (“Infringement Contentions”)
1016	Affidavit of Christopher Butler
1017	Hennessy, J., et al., “Computer Architecture A Quantitative Approach,” Morgan Kaufmann Publishers, Inc., 1990 (“Hennessy”).
1018	Joint Claim Construction Statement, <i>Collabo Innovations Inc., v. Advanced Micro Devices, Inc.</i> , 1:18-cv-552-LY (“JCCS”)
1019	Candea, G., et al., “Recursive Restartability: Turning the Reboot Sledgehammer into a Scalpel,” Proceedings Eighth Workshop on Hot Topics in Operating Systems, IEEE, May 20-22, 2001, published on IEEE Xplore: August 7, 2002. (“Candea”)
1020	U.S. Patent No. 5,784,628 to Reneris (“Reneris”)
1021	U.S. Patent No. 6,804,632 B2 to Orenstien et al. (“Orenstien”)
1022	U.S. Patent Application Publication No. 2002/0083356 to Dai (“Dai”)
1023	U.S. Patent Application Publication No. 2002/0188876 to Forbes et al. (“Forbes”)
1024	Intel Pentium III Processor/840 Developer Kit Manual (April 2001) (“Intel Pentium III Manual”)
1025	Intel Pentium 4 Processor and Intel E7205 Chipset Platform Design Guide (December 2002) (“Intel Platform Design Guide”)
1026	Summons in a Civil Action, <i>Collabo Innovations Inc., v. Advanced Micro Devices, Inc.</i> , 1:18-cv-552-LY (“Summons”)

I. Introduction

Advanced Micro Devices, Inc. (“AMD”) petitions for *inter partes* review, seeking cancelation of claims 1-7, 11, 12, 15-22, 26, 27, 30, and 31 (“challenged claims”) of U.S. Patent No. 7,930,575 to Suginaka et al. (“the ’575 patent”), assigned to Collabo Innovations, Inc. (“Collabo”). The ’575 patent is directed to a microcontroller configured to “suppress[] current consumption in time of stand-by in a power shutdown mode in which a program can proceed.” EX1001, ’575 patent, 1:13-16. But before the ’575 patent was filed, Microsoft, Intel, and others had already developed the techniques recited in the challenged claims. These developments were also extensively documented in various patents and industry publications prior to the filing of the ’575 patent. The challenged claims are therefore not patentable and should be canceled.

Petitioner’s expert, Dr. Albonesi, who has over 35 years of experience in processor architecture and chip design, explains that the microcontroller techniques recited in the challenged claims were well-known and publicly disclosed by companies in the industry well before the filing of the ’575 patent.

For example, companies like Intel filed patents, and published industry standards documents, detailing the power management techniques claimed in the ’575 patent. In U.S. Patent Pub. No. 2007/0157036 to Jahagirdar et al. (“Jahagirdar”), Intel discloses “a method and apparatus for a zero voltage

processor sleep state.” EX1005, Jahagirdar, Abstract. During a transition to the disclosed zero voltage power management state, the “operational voltage applied to the processor by the voltage regulator may be reduced to approximately zero and the state variables associated with the processor may be saved to the dedicated cache memory.” *Id.*, Abstract. Upon waking and restoring power to Jahagirdar’s CPU, it can continue where it left off in the execution stream. *Id.*, ¶¶0054, 0062.

As such, a skilled artisan would have recognized that Jahagirdar and other patents and publications disclose the techniques recited in the challenged claims of the ’575 patent. Unfortunately, however, none of these references were before the examiner during prosecution. In fact, only one reference was ever substantively considered during prosecution of the ’575 patent. And as explained below, if Jahagirdar and the other prior art cited in this petition had been before the examiner during prosecution, the ’575 patent should not have been allowed.

II. Mandatory Notices

A. Real parties-in-interest

Petitioner Advanced Micro Devices, Inc. and ATI Technologies ULC are the real parties-in-interest. ATI Technologies ULC is an indirect, wholly owned subsidiary of Advanced Micro Devices, Inc..

Inter Partes Review of U.S. Patent No. 7,930,575 B2

B. Notice of related matters

Collabo asserted the '575 patent against AMD in a litigation styled *Collabo Innovations, Inc. v. Advanced Micro Devices, Inc.*, Case No. 1:18-cv-00552-LY (W.D. TX.). This litigation is ongoing.

AMD is concurrently filing a separate petition for *inter partes* review of the '575 patent, proceeding no. IPR2019-01524.

C. Lead and back-up counsel with service information

Lead Counsel: Michael B. Ray (Reg. No. 33,997); 202.772.8569

Backup Counsel: Trent W. Merrell (Reg. No. 73,771); 202.772.8519
Michael D. Specht (Reg. No. 54,463); 202.772.8756
Jonathan Tuminaro (Reg. No. 61,327); 202.772.8967

Address: STERNE, KESSLER, GOLDSTEIN & FOX, P.L.L.C.
1100 New York Avenue, NW
Washington, DC 20005
202.371.2540 (fax)

AMD consents to service via email at:

mray-PTAB@sternekessler.com;
tmerrell-PTAB@sternekessler.com;
mspecht-PTAB@sternekessler.com;
jtuminar-PTAB@sternekessler.com; and
PTAB@sternekessler.com.

III. Grounds for Standing

AMD certifies that the '575 patent is eligible for *inter partes* review and that AMD is not barred or estopped from requesting *inter partes* review of the '575

patent.¹ The required fee is paid via online credit-card payment. The Office is authorized to charge fee deficiencies and credit overpayments to Deposit Acct. No. 19-0036 (Customer ID No. 45324).

IV. Statement of Relief Requested

AMD requests *inter partes* review and cancellation of the challenged claims based on the detailed statements presented below.

V. Identification of Challenge

AMD requests IPR on the grounds listed below. Per 37 C.F.R. §42.6(c), copies of the references are filed herewith. In support of these grounds, this petition is accompanied by a Declaration of David Albonesi Ph.D., EX1003, along with his *curriculum vitae*, EX1004. Dr. Albonesi's Declaration explains what the prior art would have conveyed to a person of ordinary skill in the art ("POSITA").

¹ AMD was served with the Complaint on September 25, 2018. See EX1026, Summons, 3.

Inter Partes Review of U.S. Patent No. 7,930,575 B2

Grounds	Challenged Claims	Type	References
1	1-3, 5-7, 11, 15, 16	§103	Jahagirdar ² and ACPI ³
2	4, 17-22, 26, 30, 31	§103	Jahagirdar, ACPI, and Saito ⁴
3	12	§103	Jahagirdar, ACPI, and Chung ⁵
4	27	§103	Jahagirdar, ACPI, Saito and Chung

As noted on the face of the patent, the earliest possible priority date of the '575 patent is September 12, 2006⁶. AMD does not concede that the '575 patent is

² U.S. Pub. No. 2007/0157036 to Jahagirdar *et al.* ("Jahagirdar"), 35 U.S.C. §102(e) (filed December 30, 2005).

³ "Advanced Configuration and Power Interface Specification," Revision 2.0, Compaq Computer Corp., Intel Corp., Microsoft Corp., Phoenix Techs. Ltd., Toshiba Corp., 35 U.S.C. §102(b) (published and publicly available at least as early as May 4, 1999).

⁴ JP Pat. Appl. No.: S58-169218 to Saito, 35 U.S.C. §102(b) (published October 5, 1983).

⁵ U.S. Pat. No. 5,390,350 to Chung *et al.* ("Chung"), 35 U.S.C. §102(b) (issued February 14, 1995).

entitled to the benefit of this date. Nonetheless, Jahagirdar, ACPI, Saito and Chung all qualify as prior art against this patent.

Note that ACPI 2.0 was released July 27, 2000, and was publically available more than one year before the '575 patent's earliest possible effective filing date. Compare ACPI with '575 patent at (30); Albonesi Dec., EX1003, ¶¶46-48. Intel filed several patent applications that refer to ACPI 2.0 and that were filed more than one year before September 12, 2006. *See e.g.*, EX1005, (22), ¶0007; EX1003, ¶¶46-50 (citing EX1022, Dai, ¶0005; EX1023, Forbes, ¶0003). The Wayback Machine also shows that the public had access to a webpage containing a link to the ACPI 2.0 document by at least May 4, 1999. *See* EX1016 (Affidavit of Christopher Butler). Thus, ACPI is prior art under 35 U.S.C. §102(b).

VI. Technical Background

The '575 patent was filed September 10, 2007, and claims priority to a Japanese application, which was filed on September 12, 2006. *See* EX1001, (22), (30). This patent is directed to a technique for suppressing current consumption in a power shutdown mode from which a program can proceed. *See id.*, 1:13-17. But

⁶ Patent Owner has conceded that the '575 patent is entitled to a priority date "as late as September 12, 2006." EX1015, Infringement Contentions, 0003.

this technique, and the benefits thereof, were known well before the '575 patent's earliest possible priority date.

A. Power management has been a longstanding concern in the computer industry.

Power management refers to a computer system's ability to conserve or manage its power consumption. Traditionally, a computer system's power management was limited to its main power switch. When the computer system was on, it consumed power; when it was off, it did not. But as more and more functionality has been packed into computer systems, their total power consumption has increased, leading to a growing demand for more sophisticated power-management techniques. EX1003, ¶40.

Power-management concerns are especially prevalent in battery-powered computing devices, such as laptops and mobile telephones. One way to increase the battery life of these devices is to reduce their power consumption. This can be achieved, for example, by reducing the operating *frequency* and the operating *voltage* of the devices' hardware. EX1003, ¶41.

But a reduction in a hardware device's operating frequency or voltage reduces not only its power consumption, but also its performance. So, effective power management must be balanced against performance. EX1003, ¶42.

To strike the appropriate balance, power-management schemes were developed. These power-management schemes can achieve energy-efficient

computation by selectively turning off (or reducing the performance of) system components when they are *idle* (or can operate at a slower speed). EX1003, ¶43. A well-known example of a power-management scheme, ubiquitously used for laptops and palmtops, is the *timeout* policy, which shuts down a component after a fixed inactivity time. *Id.*

Traditionally, power-management schemes were created and controlled by original equipment manufacturers (OEMs). In these traditional schemes, special hardware was added to monitor whether a hardware device was busy or not. If the device was not busy, embedded software would cause the device to slow down or stop its operation in order to reduce its power consumption. EX1003, ¶44.

But these OEM-based power-management schemes were problematic, because they circumvented the operating system (OS). Since the OS was not involved, if an OEM-based power-management scheme reduced the power consumption of a device, it would have to do so without impairing the operating system driver which was also attempting to utilize the device. EX1003, ¶45. But this could cause the operating system to encounter unexpected errors since a device could be powered down unexpectedly. *Id.*

B. In the mid '90s, a power-management standard, known as ACPI, was released

To address the problems with these OEM-based schemes, Microsoft—along with Intel and Toshiba—released a power-management standard that enabled the

OS to control a computing device's power consumption. EX1006, ACPI, 0015.

This power-management standard was called "Advanced Configuration and Power Interface (ACPI)." *Id.*, 0015. ACPI was dubbed as "the key element in Operating System directed configuration and Power Management." *Id.* ACPI was widely adopted and is widely implemented today in PC-based computing devices.

EX1003, ¶¶46-50.

C. The prior art discloses logic circuitry for preventing unnecessary current from flowing and for preventing the destruction of information during a shutdown event

The '575 patent acknowledges that the concepts underlying what is claimed in the patent were all previously known in the art. For example, the '575 patent admits that, at the time it was filed, microcontrollers had a "stop mode for stopping supply of a [sic] operation clock and reducing current consumption as a standby mode." EX1001, 1:19-21. The '575 patent also admits that it was known to evacuate data to an information holding circuit during "a power shutdown mode" that allows a program to proceed after a shutdown event. *See id.*, 1:40-49.

The '575 patent purports to invent a logic circuit capable of "preventing unnecessary current from flowing and a logic circuit for preventing destruction of information." *Id.*, 2:17-22. Specifically, the '575 patent purports to provide a microcontroller "capable of easily and rapidly evacuating and restoring

information necessary in proceeding with a program when realizing a standby state in which internal operation voltage supply is shutdown.” *Id.*, 2:26-31.

However, logic circuits capable of preventing unnecessary current from flowing to the CPU and for preventing the destruction of information necessary for proceeding with a program were well-known in the art.

As will be explained in greater detail below, Jahagirdar (which supports ACPI) discloses “a method and apparatus for a zero voltage processor sleep state.” EX1005, Abstract. During a transition to the disclosed zero voltage power management state, the “operational voltage applied to the processor by the voltage regulator may be reduced to approximately zero and the state variables associated with the processor may be saved to the dedicated cache memory.” *Id.*, Abstract. Upon waking and restoring power to Jahagirdar’s CPU, it can continue where it left off in the execution stream. *Id.*, ¶¶0054, 0062.

Moreover, Dr. Albonesi explains in his declaration (again with the aid of numerous support references) that the idea of using logic circuitry to prevent unnecessary current from flowing and to prevent the destruction of information during a shutdown event was a well understood technique prior to the earliest possible priority date of the ’575 patent. This understanding was reinforced by numerous publications from the relevant time period prior to the earliest possible

priority date of the '575 patent, including Jahagirdar, ACPI, and Saito. EX1003, ¶¶51-56.

VII. The '575 Patent

A. Overview

Like the prior art techniques discussed above, the '575 patent is generally directed to a technique for controlling a power shutdown process. See EX1001, Title. The '575 patent explains that its “main objective [is] to provide a microcontroller capable of easily and rapidly evacuating and restoring information necessary in proceeding with a program when realizing a standby state in which internal operation voltage supply is shutdown, a state in which an internal operation voltage is lowered, or the like.” EX1001, 2:26-31.

B. Summary of the Prosecution History

During the prosecution of the '575 patent, the United States Patent and Trademark Office (“USPTO”) issued only a single non-final office action. But, the Applicant’s response to the office action provides significant insight into what feature it believed to be novel. The office action contained a single 35 U.S.C. §102 rejection of original claims 1-26 over U.S. Patent No. 7,167,991 to Higashida. In response to the rejection, Applicant amended the independent claims and identified four features that it believed were not identically taught by Higashida. *First*, Applicant argued that Higashida’s switch is “different from the claimed power

supply unit since switch 16 is not arranged between the CPU 11 and the main power source 3;” *second*, Patent Owner argued that “Higashida fails to disclose in what timing, the information evacuated from the CPU 11 is stored in the SRAM 15;” *third*, Applicant alleged that Higashida “fails to disclose that the SRAM 15 is utilized as a stack memory and thus fails to disclose evacuating a value of a stack pointer for pointing to an evacuation place to a specific place;” and *lastly*, Applicant argued that “the cited portion of Higashida” fails to “disclose that the CPU ‘then branches a process to an address indicated by a program counter to continue program execution from a shutdown state’” EX1002, ’575 Prosecution History, 78-79.

As will be demonstrated below, each of these features was known in the art.

C. Level of Ordinary Skill in the Art

Based on the disclosure of the ’575 patent, a POSITA would have a B.S. degree in Electrical Engineering or an equivalent field as well as at least 3-5 years of academic or industry experience in computer systems architecture or computer chip design, or comparable industry experience. EX1003, ¶¶36-37. A person with less education but more relevant practical experience may also meet this standard. *Id.*

D. Claim Construction

The claim construction standard set forth in *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) applies to this proceeding. 83 Fed. Reg. No. 197, 51340 (Oct. 11, 2018); 37 C.F.R. 42.100(b). Under this standard, words in a claim are given their plain meaning, which is the meaning understood by a POSITA at the time of the alleged invention after reading the entire patent. *Phillips*, 415 F.3d at 1312-13.

Despite the parties' disagreements known from co-pending District Court litigation, because the prior art includes all elements of the challenged claims under either party's construction, Petitioner maintains that no construction may be necessary for this proceeding. Petitioner reserves the right to challenge the construction of terms in the challenged claims should a material dispute arise. At this juncture of the proceeding, Petitioner submits the following terms and constructions:⁷

⁷ Petitioner's proposed constructions, or lack thereof, for the purpose of this proceeding are not an admission that the claims are valid under 35 U.S.C. §112. Petitioner reserves the right to challenge the validity of the claims under 35 U.S.C. §112 in other proceedings and forums.

1. “Microcontroller” (claims 1, 17)

Claim Term	Collabo’s Construction	AMD’s Construction
“microcontroller”	“an integrated circuit”	plain and ordinary meaning; “a chip designed for control applications rather than data-processing that has a complete CPU, semiconductor memory sufficient for control applications, and input/output capability”

While both parties agree in the co-pending litigation that the preamble is limiting, the construction for the term “microcontroller” is in dispute. EX1018, Joint Claim Construction Statement (“JCCS”), 7-8.

The term “microcontroller” is entitled to its plain and ordinary meaning, “a chip designed for control applications rather than data-processing that has a complete CPU, semiconductor memory sufficient for control applications, and input/output capability.” EX1003, ¶¶93-94. All of these elements are required by the “microcontroller” of claim 1: a CPU, EX1001, 2:32-34, FIGs. 1, 6, 8-10, 13, 14, 19, 23, memory for execution of two microprograms, *id.*, 18:59-64, 19:18-24; *see also* EX1003, ¶¶93-101), and input-output (I/O) capability. EX1001, 16:14-34, 18:56-58, 19:8-9, Figs. 1, 6, 8-10, 13, 14, 18, 19; *see also* EX1003, ¶¶93-101.

Patent Owner argues that the term means “an integrated circuit.” EX1018, 7-8. While Petitioner applies its own construction within this petition, the prior art cited herein teaches or suggests either construction. *See* EX1005, ¶0034 (“chipset

features may be...implemented using a different number of integrated circuit chips. For example, for some embodiments, graphics and memory control capabilities may be provided using separate integrated circuit devices”), ¶0102 (“various embodiments may be applied to different types of power management configurations and/or for different types of integrated circuits”); EX1003, ¶102.

2. “CPU” (claims 1, 17)

Claim Term	Collabo’s Construction	AMD’s Construction
“CPU”	“processing units”	“processing units and associated temporary storage”

In the litigation, the parties agree that the meaning of “CPU” includes processing units, but disagree as to the associated temporary storage. EX1018, 8-9.

In the context of the ’575 patent, a POSITA would have understood the claimed CPU to mean the processing units and associated temporary storage comprised within the microcontroller. EX1003, ¶¶103-111. The claims indicate that a CPU must have processing units and associated temporary storage. The CPU “executes...microprogram[s]” and “evacuates” and “restores” “information,” which are functions corresponding both to processing units and to temporary storage. EX1001, claim 1, 1:32-35, 2:26-57.

The ’575 patent confirms that all CPUs shown in the embodiments of the patent include the processing units as well as the associated temporary storage. See *id.*, FIGs. 1, 6, 8-10, 13, 14, 19, 23 (showing microprograms μ P1 and μ P2 executed

by CPU). As such, a POSITA having read the entire patent would have understood the term “CPU” to mean processing units and associated temporary storage.

EX1003, ¶¶103-111.

Though the parties agree that the meaning of “CPU” includes processing units, but disagree as to the associated temporary storage, the same prior art disclosures teach or suggest either construction.

3. “Power supply unit” (claims 1, 17)

Claim Term	Collabo’s Construction	AMD’s Construction
“power supply unit”	“Power switch switching power supplied by the power supply device.”	112¶6 applies: Structure: “voltage regulator” Function: “supplying power”

This term should be construed under 35 U.S.C. §112¶6 for failing to connote structure to a POSITA. Although “means” is not recited in the claim, the rebuttable presumption should be overcome, because in the context of claims 1 and 17, which recite several power supply terms including a “power supply device,” “power supply unit” and “power supply control unit,” the term “power supply unit” lacks sufficiently definite structure. EX1003, ¶112. This claim term uses a generic placeholder or nonce term “unit” as a substitute for “means” and is also modified by functional language. MPEP 2181(I).

The function set forth in claims 1 and 17 is “*supplying power.*” EX1001, 18:48-49, 20:51-52. Additionally, the structure for the “power supply unit” should

be construed to be “a voltage regulator” (or its equivalents), because this is the only structure identified in the specification as corresponding to the claimed “power supply unit.” *Id.*, 8:51-52, 14:47-62 (“power supply unit[s]” which are “configured by a general voltage regulator”); EX1003, ¶¶112-116.

Alternatively, if the Board applies Patent Owner’s construction for the term “power supply unit” as meaning a “power switch switching power supplied by the power supply device.” EX1018, 10-12. The prior art cited herein also teaches or suggests a “power switch switching power supplied by the power supply device” under this construction. EX1003, ¶117. For example, Jahagirdar teaches that its voltage regulator may be used to “shut off” the operating voltage for the CPU, see EX1005, ¶¶0031-0033, 0061; EX1003, ¶117.

Thus, the prior art cited herein teaches or suggests this term under either construction. EX1003, ¶117.

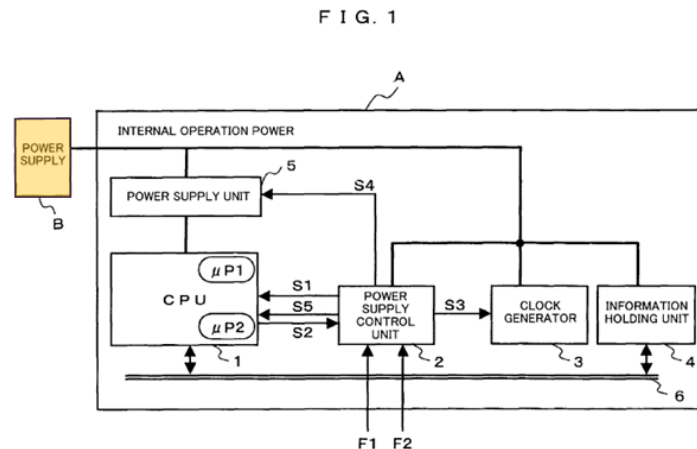
4. “Power supply device” (claims 1, 17)

Claim Term	Collabo’s Construction	AMD’s Construction
“power supply device”	“power supply”	“external power supply”

The construction for the term “power supply device” is in dispute. EX1018, 9-10.

Neither of the independent claims provides any insight as to the proper scope of this term. However, the ’575 figures depict the power supply device as

being external to the microcontroller. See EX1001, FIGs. 1, 6, 8-10, 13, 14, 19, 23. For example, as illustrated in Figure 1 below, power supply B (shaded orange) is external to microcontroller A.



EX1001, FIG. 1 (annotated).

The specification consistently illustrates in all nine block-level diagrams that the power supply ‘B’ is external to microcontroller ‘A.’ EX1001, FIGs. 1, 6, 8-10, 13, 14, 19, 23, 8:44-51. This view is consistent with the prosecution history in which the Applicant amended the claims by removing the separate requirement that a “power supply device” be a component of (i.e., within) the “microcontroller.” EX1002, 67, 77-78. Therefore it is to be understood that the power supply device is external to the microcontroller. EX1003, ¶¶118-121.

The prior art cited herein teaches or suggests an external power source and thereby teaches or suggests this term under either construction. EX1003, ¶122.

5. “Power supply control unit” (claims 1, 17)

AMD contended in the co-pending district court proceeding that this term should be interpreted under 35 U.S.C. § 112¶6. EX1018, 12-13. The function is controlling a power supply. EX1018, 12-13; EX1001, 18:50-51 (“controlling the CPU and the power supply unit”); 20:53-54. However, AMD also contended in the co-pending district court proceeding that there is insufficient structure for this term, and thus it is indefinite under 35 U.S.C. § 112. EX1018, 12-13.

Given that AMD cannot raise indefiniteness challenges in this proceeding, solely for the purposes of this proceeding, AMD adopts Patent Owner’s interpretation from the district court proceeding, in which Patent Owner argues for plain and ordinary meaning and states that the term connotes structure to a POSITA. EX1018, 12-13. Based on this, AMD understands Patent Owner’s construction to be logic/circuitry for controlling the CPU and power supply unit.

As shown below, the prior art teaches or suggests the claimed “power supply control unit” under Collabo’s construction.

6. “Information holding unit” (claims 1, 17)

Claim Term	Collabo’s Construction	AMD’s Construction
“information holding unit”	“No construction necessary;” memory	35 U.S.C. §112¶6 applies: Structure: “built-in RAM external to the CPU” Function: “holding information”

This term should be construed under 35 U.S.C. §112¶6 for failing to connote structure to a POSITA. Although “means” is not recited in the claim, the rebuttable presumption should be overcome, because the term “information holding unit” lacks sufficiently definite structure. EX1003, ¶124. This claim term uses a generic placeholder or nonce term “unit” as a substitute for “means” and is also modified by functional language. MPEP 2181(I).

The function set forth in claims 1 and 17 is “*holding information.*” EX1001, 18:52-54, 20:56-58. Additionally, the structure for the “power supply unit” should be construed to be “built-in RAM external to the CPU” (or its equivalents), because this is the only structure identified in the specification as corresponding to the claimed “information holding unit.” *Id.*, FIG. 1, 8:39-42 (“The microcontroller A includes . . . an information holding unit (built-in RAM) 4.”); EX1003, ¶¶125-129 (explaining that the information holding unit is external to the CPU to store evacuated information).

Alternatively, if the Board determines that 35 U.S.C. §112¶6 does not apply, the same prior art disclosures cited herein teach or suggest a “memory,” as per Patent Owner’s interpretation. EX1018, 17; EX1003, ¶129.

The prior art cited herein teaches or suggests this term under either construction. EX1003, ¶129.

7. “Restores the information necessary in proceeding with the program and evacuated in the information holding unit in time of power shutdown” (claim 1)

Claim Term	Collabo’s Construction	AMD’s Construction
“restores the information necessary in proceeding with the program and evacuated in the information holding unit in time of power shutdown”	“restores the information necessary to proceed with the program, the information was evacuated from the CPU in the information holding unit in the power shut down process.”	“restores from the information holding unit the information necessary to proceed with the program, the information was stored in the information holding unit during power shutdown”

The parties disagree as to whether an information holding unit through which the evacuated information simply passes satisfies the present claim term even if the information is not stored in that unit when power to the CPU is shut down. EX1018, 16-17.

The plain language of the claim requires that information be evacuated to the information holding unit and then restored. EX1001, claim 1. The specification of the ’575 patent discloses that “[u]pon receiving the restoration request signal S5 from the power supply control unit 2, the CPU 1 activates a power supply

restoration microprogram μ P2, and restores the information necessary in proceeding with the program that had been evacuated to the information holding unit 4 in time of power shutdown to the CPU.” *Id.*, 9:19-28, 10:18-23, 2:58-3:6, 4:65-5:9. EX1003, ¶¶130-134.

Because the prior art cited herein teaches or suggests restoring information that was stored in the information holding unit during power shut down, those same disclosures also teach or suggest Patent Owner’s construction. EX1003, ¶135.

8. “The power supply control unit limits a period in which shutdown is possible” (claims 3, 19)

AMD contended in the co-pending district court proceeding that this term should be interpreted under 35 U.S.C. §112¶6. EX1018, 13-15. The function is limiting a period in which shutdown is possible. EX1018, 13-15; EX1001, 19:31-33. However, AMD also contended in the co-pending district court proceeding that there is insufficient structure for this term, and thus it is indefinite under 35 U.S.C. § 112. EX1018, 13-15.

As stated above, given that AMD cannot raise indefiniteness challenges in this proceeding, solely for the purposes of this proceeding, AMD adopts Patent Owner’s interpretation from the district court proceeding, in which Patent Owner argues for plain and ordinary meaning and states that the term connotes structure to a POSITA. EX1018, 12-14. Based on this, AMD understands Patent Owner’s

construction to be a “power supply control unit [that] sets the time in which shutdown is possible.” EX1018, 13-14; See Section VII.D.5.

As shown below, the prior art teaches or suggests Collabo’s construction, because Jahagirdar’s teachings are consistent with Patent Owner’s Infringement Contentions where Patent Owner asserts that waiting for a period of time before flushing CPU caches reads on this claim. EX1015, 0031, 0112-0113.

9. “The plurality of microprograms”/ “a program for resetting processes, and the CPU selects one of the plurality of microprograms in an alternative way” (Claims 7, 22)

AMD contended in the co-pending district court proceeding that this term is indefinite. EX1018, 17-18. Given that AMD cannot raise indefiniteness challenges in this proceeding, solely for the purposes of this proceeding⁸, AMD adopts Patent Owner’s interpretation of this term from the district court proceeding, which is “a program for resetting processes, and the CPU selects one of [a microprogram for restoring the evacuated information necessary in proceeding with the program and proceeding with the program or a microprogram for restoring a set value without using the evacuated information necessary in proceeding with the program] ... in an alternative way.” EX1018, 17-19.

⁸ Petitioner reserves the right to challenge whether this term is indefinite under 35 U.S.C. §112 in this and other jurisdictions.

10. “A circuit for loading an inverted value to a register in a circuit portion in a power shutdown range” (claims 12, 27)

AMD contended in the co-pending district court proceeding that this term is indefinite. EX1018, 19-21. Given that AMD cannot raise indefiniteness challenges in this proceeding, solely for the purposes of this proceeding⁹, AMD adopts Patent Owner’s interpretation of this term from the district court proceeding, which is “a circuit for loading opposite Boolean values to a register in a circuit portion, the opposite Boolean values being in a range for power shutdown” EX1018, 19-21.

VIII. Grounds of Rejection

Claims 1-7, 11, 12, 15-18-22, 26, 27, 30, and 31 are unpatentable for at least the reasons set forth below.

A. Ground 1: Claims 1-3, 5-7, 11, 15, and 16 are obvious over Jahagirdar and ACPI

As will be described in greater detail below when applied to claims 1-3, 5-7, 11, 15, and 16, Jahagirdar discloses a “a method and apparatus for a zero voltage processor sleep state.” EX1005, Abstract. During a transition to the disclosed zero voltage power management state, the “operational voltage applied to the processor by the voltage regulator may be reduced to approximately zero and the state

⁹ Petitioner reserves the right to challenge whether this term is indefinite under 35 U.S.C. §112 in this and other jurisdictions.

variables associated with the processor may be saved to the dedicated cache memory.” *Id.* Upon waking and restoring power to Jahagirdar’s CPU, it can continue where it left off in the execution stream. *Id.*, ¶¶0054, 0062; EX1003, ¶¶64-65.

ACPI was developed “to establish industry common interfaces enabling robust operating system (OS)-directed motherboard device configuration and power management of both devices and entire systems.” EX1006, 0015. ACPI further describes “OSPM run-time aspects of managing the processor’s performance, power consumption, and other controls,” including processor “power states: C0, C1, C2, C3...Cn.” *Id.*, 0225.

1. Motivation to combine: Jahagirdar suggests the proposed combination.

Jahagirdar and ACPI are directed to the same purpose—controlling the power consumption of a processor by controlling its supply voltage. See EX1005, abstract; EX1006, §§3.5, 8. Jahagirdar expressly supports and uses the power-management concepts disclosed in ACPI, therefore combining the teachings of these references would have yielded no unexpected results. *See KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 414-418 (2007). As such, no specific motivation to combine is necessary. *See id.*

That said, a POSITA would have been motivated to combine Jahagirdar with ACPI. EX1003, ¶¶140-143. Again, ACPI is a widely accepted industry standard

that allows the OS to manage the power consumption of a processor. *See* EX1006, §§3.5, 8, 9. Building off these concepts, Jahagirdar explicitly references ACPI and then discloses further techniques to manage the power consumption of a processor. *See*, EX1005, ¶¶0007-0008, 0020, 0039. Indeed, Intel is the assignee of Jahagirdar and is also one of ACPI's promoters. *See* EX1005, (73); EX1006, 0015.

Given these connections between these references, a POSITA would have naturally looked to combine Jahagirdar with ACPI in order to fill in any gaps in the Jahagirdar disclosure. *See KSR*, 550 U.S. at 420 (“[I]n many cases a person of ordinary skill [in the art] will be able to fit the teachings of multiple patents together like pieces of a puzzle.”) Jahagirdar discloses the use and implementation of ACPI but does not include within its four corners all details of the ACPI standard. Thus, a POSITA would have been specifically directed to ACPI to understand those implementation details and the full capabilities under the supported ACPI standard. EX1003, ¶142. For example, ACPI provides additional details about the “processor sleeping states [C-states] where the processor consumes less power and dissipates less heat.” EX1006, 0225. Each C-state possesses specific entry and exit conditions and procedures, and the Petition refers to ACPI's details for transitioning between the C-states to better understand Jahagirdar's teachings. *Id.*, 0037, 0229. ACPI further discloses additional details about Jahagirdar's clock stop control signal (STPCLK#), namely that the command

may be used to implement “eight levels of clock control” to emulate different frequency settings. *Id.*, 0228.

Again, one of the promoters of ACPI was Microsoft, and Microsoft Windows was ubiquitous in the late '90s and early 2000s. EX1003, ¶143; *see also* EX1014, PC Magazine Report (showing that Windows had nearly 80% of the desktop market in 1997 and was projected to have over 90% of the market by 2001). As such, a POSITA would naturally look to combine Jahagirdar with ACPI. EX1003, ¶¶140-143.

2. Claim 1: The combination of Jahagirdar and ACPI renders this claim obvious.

a) Preamble of claim 1¹⁰

As set forth in Section VII.D.1 above, the term “microcontroller” means a chip designed for control applications rather than data-processing that has a complete CPU, semiconductor memory sufficient for control applications, and input/output capability. EX1003, ¶145.

Jahagirdar’s system 200 (i.e., microcontroller) comprises a processor 205 (i.e., CPU), a memory 220 which may comprise one or more of random access memory (RAM), flash memory and/or another type of memory (i.e.,

¹⁰ Both Patent Owner and Petitioner agree that the preamble is limiting. EX1018, 7.

semiconductor memory), and an input/output (I/O) control hub 225 (i.e., input/output capability). EX1005, ¶0025.

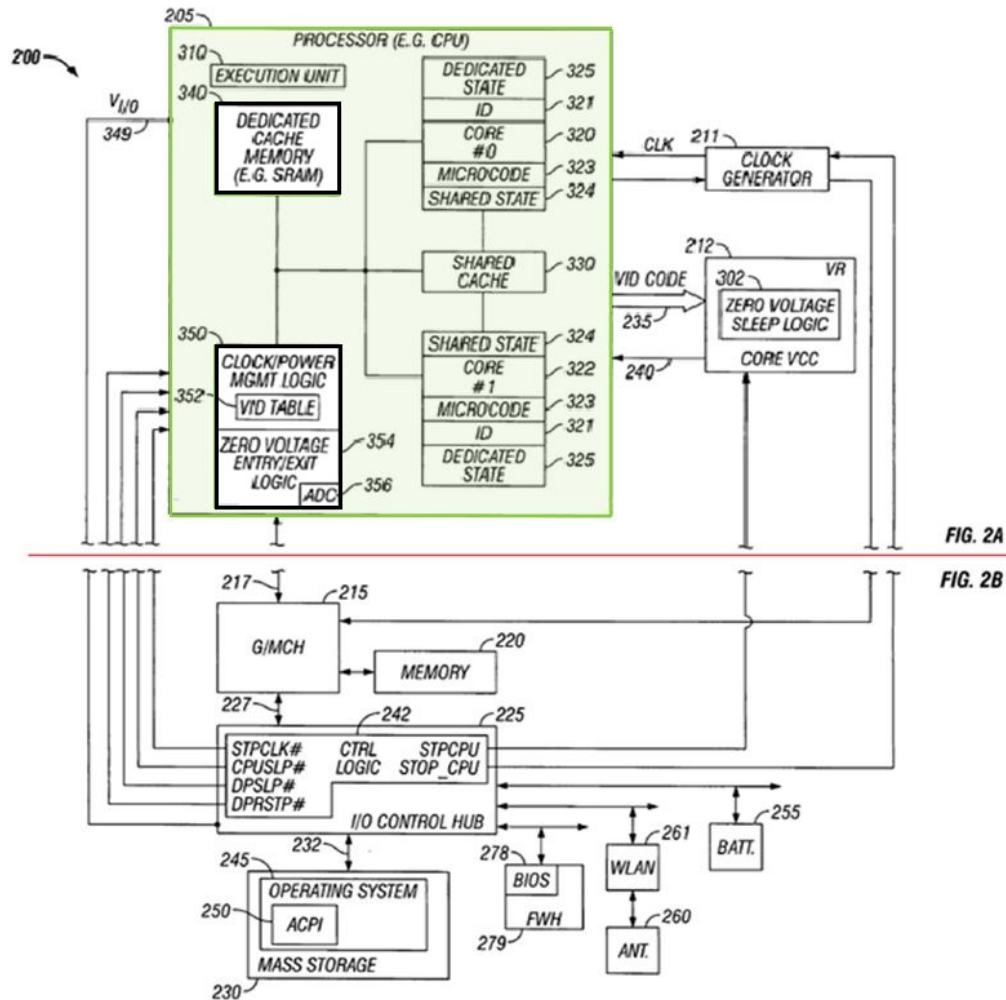
Jahagirdar's components can be located on the same chip. *Id.*, ¶0102 ("It will be appreciated that...various embodiments may be applied to different types of power management configurations and/or for different types of integrated circuits"); EX1003, ¶147. Furthermore, Jahagirdar teaches that the processor could be an embedded processor, which is synonymous with a microcontroller. EX1005, ¶0027; EX1003, ¶147.

Thus, a POSITA would have understood that the disclosed system 200 teaches or suggests the preamble. EX1003, ¶¶145-148.

b) Limitation [1.A]¹¹

As illustrated in Figures 2A-B below, Jahagirdar discloses "a central processing unit (CPU) 205 [shaded green]," (EX1005, ¶0050), with each core storing microcode 323 thereon and having access to a shared cache 330. *Id.*, ¶¶0057-0058.

¹¹ The complete claim language for each challenged claim can be found in Appendix A.



EX1005, FIGs. 2A, 2B (annotated to show processor 205 shaded in green, excluding dedicated cache memory 340 and clock/power management unit 350).

Thus, Jahagirdar discloses element [1.A]. EX1003, ¶149.

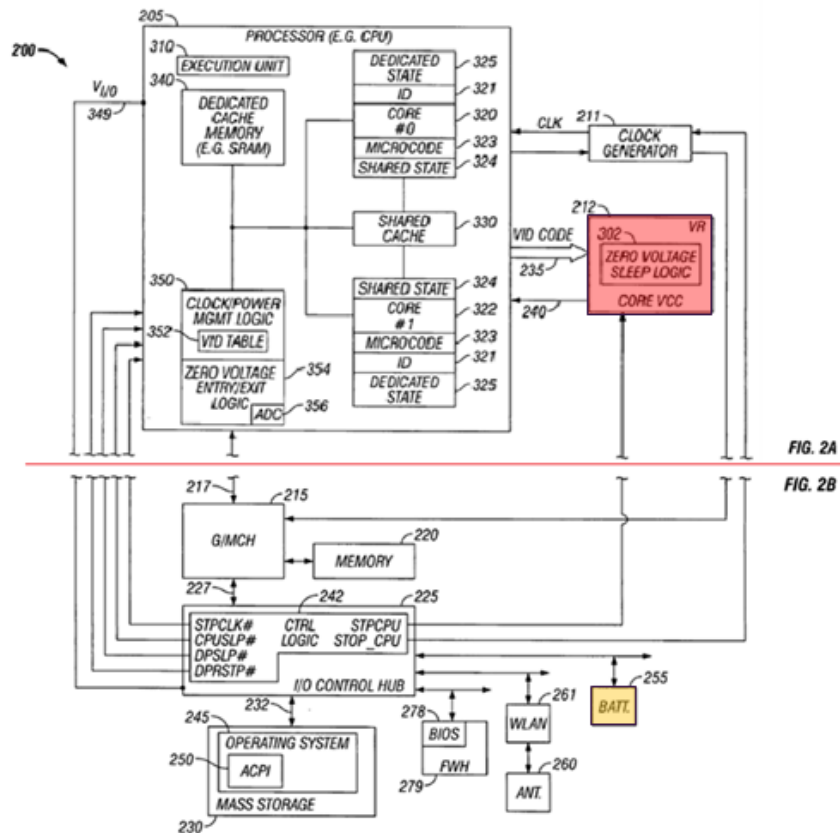
c) Limitation [1.B]

As explained in Section VII.D.3, the term “power supply unit” should be construed under 35 U.S.C. §112(f). The corresponding structure of the “power supply unit” is the voltage regulator disclosed at EX1001, 12:43-67, 20:43-44, and equivalents thereof. EX1003, ¶150. The function of the “power supply unit” is

“supplying power.” Jahagirdar’s voltage regulator teaches or suggests the structure and function of the recited “power supply unit.” EX1005, ¶0033; EX1003, ¶150.

Specifically, Jahagirdar discloses a voltage regulator (i.e., power supply unit) arranged between the CPU and an external battery 255 (i.e., power supply device) for supplying power to the CPU. EX1003, ¶151.

For example, Jahagirdar teaches that its voltage regulator 212 “provides a supply operating voltage to [CPU] 205 and may be in accordance with a version of the Intel Mobile Voltage Positioning (IMVP) specification such as the IMPV-6 specification.” EX1005, ¶0033; EX1003, ¶152. As illustrated in Figures 2A-B, reproduced and annotated below, voltage regulator 212 (shaded red) is located between CPU 205 and a power supply device battery 255 (shaded orange). EX1005, FIGs. 2A-B, ¶0042 (“a battery or battery connector 255 may be included to provide power to operate the system 200.”)



EX1005, FIGs. 2A-2B (annotated).

As illustrated in Figure 2A, line 240 extends from the voltage regulator to the CPU and represents the “operating voltage 240” for the CPU. EX1005, ¶10036.

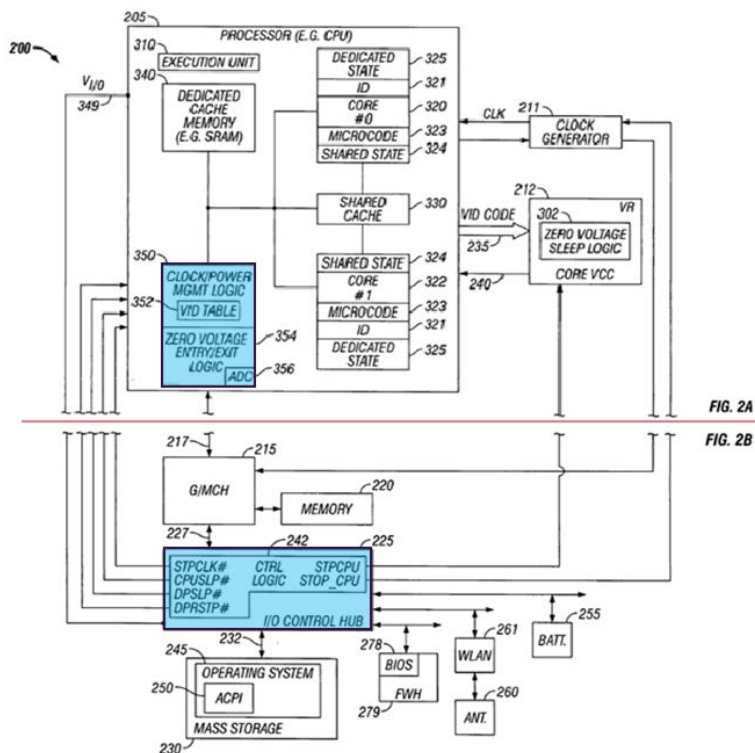
Jahagirdar’s use of a voltage regulator for supplying operating voltage from a power source to the CPU is consistent with conventional practices and was known in the art. EX1003, ¶153. That is, a voltage regulator was conventionally placed between a power source and the CPU, to regulate the amount of voltage that is passed on to the CPU as an operating voltage. *Id.* Accordingly, Jahagirdar discloses claim element [1.B]. *Id.*

To the extent Patent Owner argues that the power supply unit and the CPU must reside on the same semiconductor substrate (“chip”), although not shown in Figures 2A-2B, Jahagirdar is configured to operate as an integrated circuit, meaning, the voltage regulator (i.e., power supply unit) and CPU can be located on the same chip. EX1003, ¶154.

For instance, Jahagirdar first introduces its processor as “an integrated circuit device.” EX1005, ¶0020. Jahagirdar also teaches that “the voltage regulator may be integrated with another component of the system 200...[and] that the voltage regulator may or may not be integrated with the CPU dependent upon design considerations.” *Id.*, ¶0033; EX1003, ¶155. Finally, Jahagirdar explains that its various embodiments may be applied to “different types of integrated circuits.” EX1005, ¶0102. In sum, a POSITA would have understood that Jahagirdar’s system could be implemented on a single chip. EX1003, ¶¶150-155.

d) Limitation [1.C]

As illustrated in Figures 2A-2B below Jahagirdar’s I/O control hub, clock/power management unit 350, and zero voltage entry/exit logic 354 together teach or suggest the recited “power supply control unit.” EX1005, ¶0033.



EX1005, FIGs. 2A-2B (annotated).

In the illustrated embodiment, Jahagirdar’s clock/power management unit 350 and zero voltage entry/exit logic 354 are located on CPU 205, however, it would have been obvious to a POSITA that unit 350 and logic 354 could reside at control hub 225 or some other location outside CPU 205. EX1003, ¶156. Such a modification would have been straightforward for a POSITA to implement and would have yielded no unexpected results. *Id.*

The I/O control hub 225 “includes power management state control logic 242, alternatively referred to herein as C-state control logic,” which controls “aspects of the transitions between some power management and/or normal operational states associated with the processor 205, either autonomously or in

response to operating system or other software or hardware events.” EX1005, ¶0035. For example, Jahagirdar discloses that “for Intel® architecture processors for which at least active mode and power management states referred to as C0¹², C1, C2 and C4 are supported, the power management state control logic 242 may at least partially control transitions between at least a subset of these states using one or more of a stop clock (STPCLK#), processor sleep (SLP#), deep sleep (DPSLP#), deeper stop (DPRSTP#), and/or stop processor (STPCPU#) signals.” *Id.*

Jahagirdar states that “the deeper stop signal (DPRSTP#) is deasserted [at the I/O control hub] (circle 502) which is detected by the CPU 205 and the zero voltage entry/exit logic 354 of the clock/power management logic 350 such that the low frequency mode (LFM) VID (voltage identification) is sent to the voltage regulator 212. This indicates to the voltage regulator 212 to drive the core operational voltage [applied to the CPU] back up to the required VID.” *Id.*, ¶0078. Thus, both the CPU and voltage regulator are controlled and powered on/off by the I/O control hub’s deeper stop signal, clock/power management unit 350’s VID

¹² ACPI explains that the “C” states are “processor sleeping states where the processor consumes less power and dissipates less heat.” EX1006, 0225. Each C-state possesses specific entry and exit conditions and procedures. *Id.*, 0037, 0229.

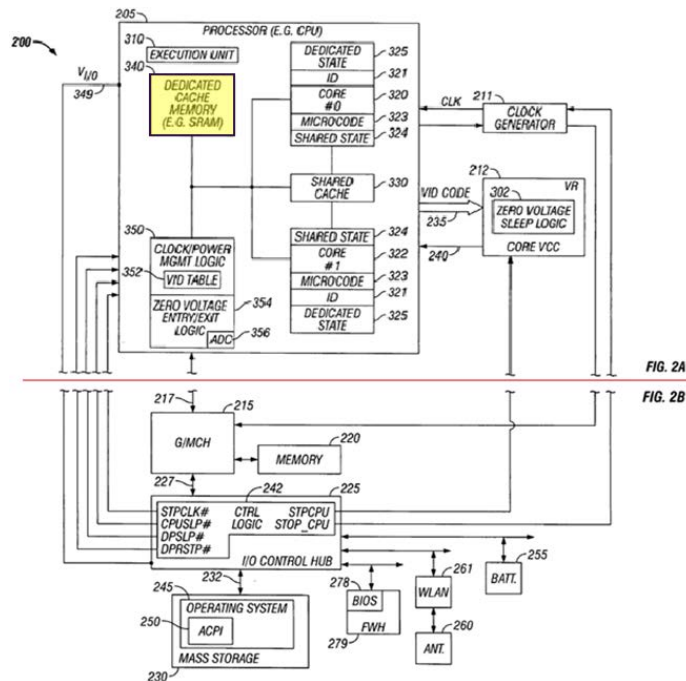
signal sent in response to the deeper stop signal being deasserted, and zero voltage entry/exit logic 354. EX1003, ¶¶156-158.

Accordingly, Jahagirdar discloses claim element [1.C]. EX1003, ¶¶156-159.

e) Limitation [1.D]

As explained in Section VII.D.6, the term “information holding unit” should be construed 35 U.S.C. §112¶6. The corresponding structure of the “information holding unit” is the built-in RAM external to the CPU disclosed at EX1001, FIG. 1, 8:39-44, and equivalents thereof. EX1003, ¶160. The function of the “information holding unit” is “holding information.” Jahagirdar’s sleep state SRAM or dedicated cache memory 340 teaches or suggests the structure and function of the recited “information holding unit.” EX1005, ¶0028; EX1003, ¶160.

Jahagirdar’s SRAM 340 (shaded yellow) may be built into the processor’s chip, as illustrated in Figure 2A, or “packaged within the same housing as the processor chip” (i.e., external to the CPU). See EX1005, ¶0028. Thus, a POSITA would have understood that SRAM 340 could reside outside CPU 205. EX1003, ¶161.



EX1005, FIGs. 2A-2B (annotated).

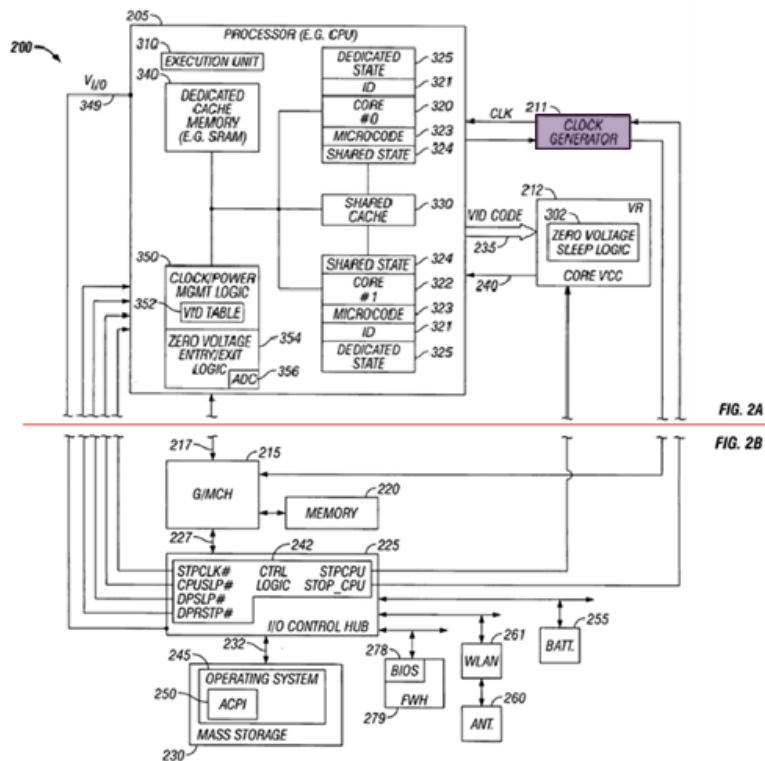
SRAM 340 stores critical state information flushed (i.e., evacuated) from the CPU such that, upon reset, the CPU can continue where it left off in the execution stream (i.e., information being necessary in proceeding with a program). EX1005, ¶¶0028, 0050, 0053-0054, 0056; EX1003, ¶162.

The critical state variables that are flushed from the CPU, includes “state variables associated with the architectural, micro-architectural, debug state, and/or similar state variables associated with that processor.” EX1005, ¶¶0020, 0050, 0072; *see also* EX1003, ¶163 (citing EX1017, Hennessy, 201 “the state includes program counter (PC)”).

Therefore, Jahagirdar discloses claim element [1.D]. EX1003, ¶¶160-164.

f) Limitation [1.E]

As illustrated in Figures 2A-2B, Jahagirdar's system 200 includes a clock generator (shaded purple).



EX1005, FIGs. 2A-2B (annotated).

Accordingly, Jahagirdar discloses claim element [1.E]. EX1003, ¶165.

To the extent Patent Owner argues that the clock generator and the CPU must reside on the same chip, although not shown in Figures 2A-2B, Jahagirdar is configured to operate as an integrated circuit, meaning, clock generator and CPU can be located on the same chip. *See supra* Section VIII.A.2.c; EX1003, ¶166. For instance, Jahagirdar first introduces its processor as “an integrated circuit device,”

EX1005, ¶0020, and Jahagirdar explains that its various embodiments may be applied to “different types of integrated circuits.” *Id.*, ¶0102; EX1003, ¶¶165-166.

g) Limitation [1.F]

Jahagirdar’s I/O control hub outputs multiple shutdown request signals (i.e., signals that control transitions between at least a subset of the C-states) at different stages as the CPU transitions between various known C-states to conserve power and reduce thermal load (i.e., power shutdown factors). *See* EX1005, ¶¶0035, 0045-0052; EX1003, ¶167.

Jahagirdar discloses an event where its CPU “may be transitioned to a lower power state... in response to firmware, such as microcode, or software, such as the operating system 245, or even ACPI software in some cases, executing a HALT or MWAIT instruction (not shown), the processor 205 may transition to the C1 or Auto-HALT state.”¹³ EX1005, ¶0045. Jahagirdar further explains that in the C1 or Auto-HALT state, “portions of the processor 205 circuitry may be powered down and local clocks may be gated.” *Id.*, ¶0045; EX1003, ¶168.

Jahagirdar further teaches that the power management state control logic 242 may control transitions between at least a subset of the C-states using one or more

¹³ Throughout this document, underlining for emphasis has been added unless indicated otherwise.

of a stop clock (STPCLK#)¹⁴, processor sleep (SLP#), deep sleep (DPSLP#), deeper stop (DPRSTP#), and/or stop processor (STPCPU#) signals.” EX1005, ¶0035.

It would have been obvious to a POSITA that the recited “power shutdown factor” includes events triggered to balance performance, power consumption, battery life, thermal requirements, and noise-level requirements. *See* EX1006, §§ 8 (“Processor Control”), 12 (“Thermal Management”); *see also* EX1005, ¶0045 (explaining that the CPU may be transitioned to a lower power C-state to “conserve power and/or reduce thermal load”); EX1003, ¶170.

Accordingly, Jahagirdar discloses claim element [1.F]. EX1003, ¶¶167-171.

h) Limitations [1.G.a-c]

Jahagirdar’s CPU executes microcode 323 in response to receiving the shutdown request signals from the I/O control hub. *Supra* Section VIII.A.2.g.

For example, as illustrated in Figure 4 below, the series of state transitions (e.g., circles 404, 406, and 408) are “directed by the microcode 323 of the CPU cores 320 and 322 of CPU 205” to transition to a lower power state, in response to

¹⁴ This mapping is consistent with Patent Owner’s Infringement Contentions where Patent Owner argues that a STPCLK# signal is a shutdown request signal. EX1015, 0013-0016.

(i.e., upon receiving) a shutdown request signal from the I/O control hub. EX1005, ¶¶0045, 0065; EX1003, ¶¶172-173.

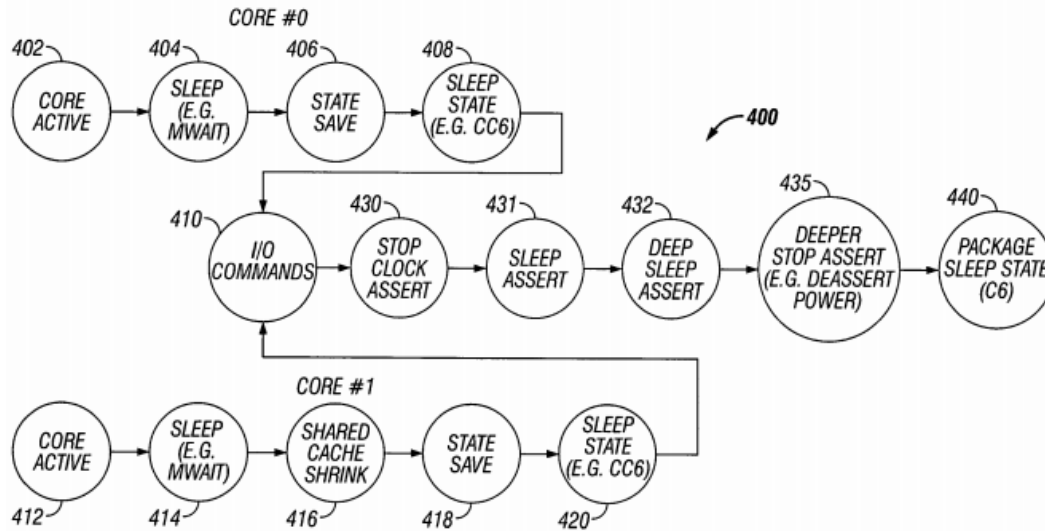


FIG. 4

EX1005, FIG. 4.

Thus, Jahagirdar discloses claim element [1.G.a]. EX1003, ¶¶174.

With respect to limitation [1.G.b], Jahagirdar’s CPU flushes (i.e., evacuates) critical state variables associated with the architectural, micro-architectural, debug state, and/or similar state variables (i.e., information necessary in proceeding with the program) into dedicated cache memory 340 (i.e., the information holding unit). See EX1005, ¶¶0020, 0068-0069; EX1003, ¶175.

Jahagirdar teaches that at “this point, all the required state of the CPU 205 has been saved or flushed from the CPU 205,” such that, upon reset, the CPU can continue where it left off in the execution stream. EX1005, ¶¶0054, 0068, 0072.

Accordingly, Jahagirdar discloses claim element [1.G.b]. EX1003, ¶¶175-176.

With respect to limitation [1.G.c], after microcode operations have been performed, Jahagirdar transfers control of the data flow to the zero voltage entry/exit logic 354. *See* EX1005, ¶0071. Jahagirdar further explains that the sleep states are not entered, until the required CPU state information has been saved:

At this point, all the required state of the CPU 205 has been saved or flushed from the CPU 205. The zero voltage entry/exit logic 354 of the clock/power management logic 350 then initiates an external platform level entry sequence (e.g. a C6) sequence by performing an I/O register read from the [Control Hub] 225. In one embodiment, this may be the ACPI defined method of entering CPU “C” states.

EX1005, ¶0072; *see also id.*, ¶¶0068-0071.

In view of the foregoing, it would have been obvious to a POSITA that clock/power management logic 350 would have initiated the sleep state transition in response to an indication that state data had been saved. EX1003, ¶177. This occurs during the state transition between 406 and 408 for Core #0, and 418 and 420 for Core #1, in Fig. 4. EX1003, ¶177. And it would have been obvious to a POSITA that the indication (i.e., evacuation completed signal) could be in the form of an internal signal or setting a flag. *Id.* Thus, Jahagirdar renders limitation [1.G.c] obvious. EX1003, ¶177.

Accordingly, Jahagirdar teaches or suggests claim elements [1.G.a-c]. EX1003, ¶¶177-178.

i) Limitation [1.H]

As explained in Section VIII.A.2.d, when Jahagirdar's I/O control hub outputs a DPRSTP# command, the clock/power management unit 350 responds by switching the CPU VID signal to a zero voltage level (i.e., a power shutdown control signal) and thereby removes power to the CPU. EX1005, ¶0074; EX1003, ¶179. For example, Jahagirdar teaches that when ACPI software 250 detects no pending interrupts, it causes the "ICH 225 to assert one or more power management-related signals such as the exemplary Deeper Stop (DPRSTP#)... and causes clock/power management logic 350...to initiate a low frequency mode (LFM)," from which other low power modes can be achieved. EX1005, ¶0048.

As explained in Section VIII.A.2.h, after microcode operations have been performed and after the required state data of the CPU 205 has been saved or flushed from the CPU 205, Jahagirdar's system initiates an external platform level entry sequence (e.g. a C6) by performing an I/O register read at the control hub. See EX1005, ¶¶0071-0072; EX1003, ¶180. Jahagirdar explains that this process is "the ACPI defined method of entering CPU 'C' states."¹⁵ EX1005, ¶0072.

¹⁵ ACPI explains that the "C" states are "processor sleeping states where the processor consumes less power and dissipates less heat." EX1006, 0225. Each C-state possesses specific entry and exit conditions and procedures. *Id.*, 0037, 0229.

Control Hub 225 then issues specific power state control commands to direct transitions between C-states. *See id.*, ¶0035. For example, when the Control Hub issues a DPRSTP# command, the CPU VID signal switches to a zero voltage level “to tell the voltage regulator 212 to remove power and that it is safe to do so.” *Id.*, ¶0074.

As described above, zero voltage entry/exit logic 354 operates in conjunction with zero voltage sleep state logic 302 of voltage regulator 212. *See id.*, ¶¶0049-0055; EX1003, ¶183. And based on the disclosure of CPU VIDs, a POSITA would have understood that entry/exit logic 354 sends a CPU VID command (i.e., power shutdown control signal) corresponding to zero voltage sleep state logic 302 to shutdown the supply of power to the CPU. EX1003, ¶¶182-184. In this way, power to the CPU is deasserted (i.e., shutdown). EX1005, ¶0074.

Accordingly, Jahagirdar discloses claim element [1.H]. EX1003, ¶¶179-184.

j) Limitation [1.I]

As explained in Section VIII.A.2.i, when the Control Hub issues a DPRSTP# command, as illustrated in Figure 4, circle 435, below, the clock/power management unit 350 responds by switching the CPU VID signal to a zero voltage level “to tell the voltage regulator 212 to remove power and that it is safe to do so.” EX1005, ¶0074. In this way, power to the CPU is deasserted (i.e., shutdown) and

voltage regulator 212 adjusts the operating voltage to zero (i.e., the power supply to the CPU is shut down). *Id.*, ¶¶0074, 0076; EX1003, ¶185.

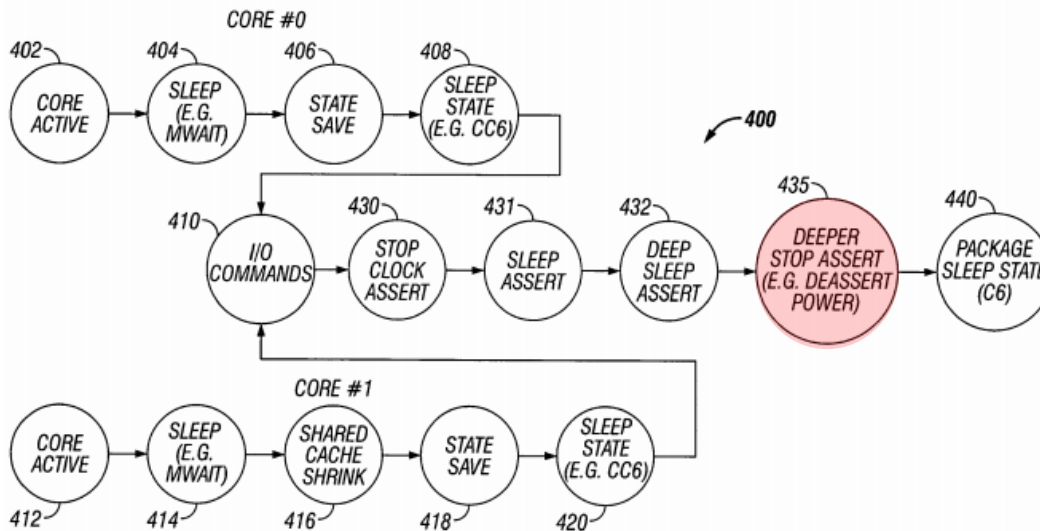


FIG. 4

EX1005, FIG. 4 (annotated).

Therefore, Jahagirdar discloses claim element [1.I]. EX1003, ¶¶185-186.

k) Limitation [1.J]

Jahagirdar teaches or suggests receiving an evacuation complete signal and that the system enters the CPU C-states. *See supra* Section VIII.A.2.h; EX1005, ¶0072. As illustrated in Figure 4 below, I/O commands 410 may be issued from control hub 225, including commands such as STOP_CPU (i.e., clock stop control signal) to halt the clock signal to the CPU. EX1005, ¶0047 (“STOP_CPU signal may be asserted by the [control hub 225] and received by the clock generator 211 to cause the clock generator to halt the clock signal CLK to the CPU 205”), ¶0073; EX1003, ¶187.

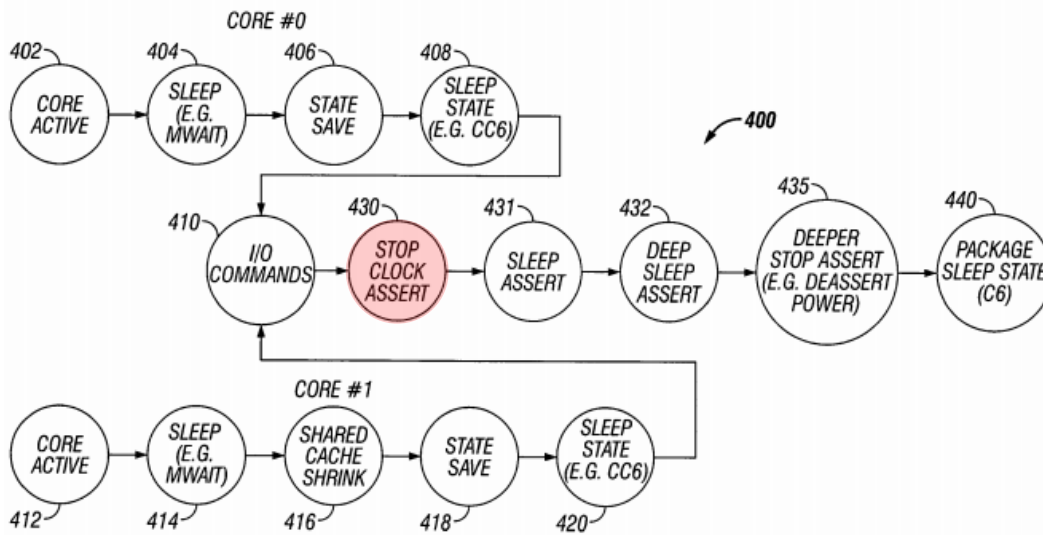


FIG. 4

EX1005, FIG. 4 (annotated).

Therefore, Jahagirdar discloses claim element [1.J]. EX1003, ¶¶187-188.

l) Limitation [1.K]

Jahagirdar teaches an exit sequence from the zero voltage processor sleep state that typically occurs when an interrupt event (i.e., a power supply restoration factor)¹⁶ necessitates waking up the CPU, which causes a low frequency mode VID (i.e., power supply control signal) to be sent to the voltage regulator (i.e., power supply unit). EX1005, ¶¶0077-0078; EX1003, ¶189.

¹⁶ This mapping is consistent with Patent Owner's Infringement Contentions where Patent Owner maps the restoration factor to C-state exit events, including interrupts. EX1015, 0020-0021.

Jahagirdar's exit sequence "may be considered the reverse of what happens during the previously-described entry phase." EX1005, ¶0077. At circle 502 of Figure 5 below, Jahagirdar's control hub de-asserts the deeper stop signal (DPRSTP#) "which is detected by the CPU 205 and the zero voltage entry/exit logic 354 of the clock/power management logic 350 such that the low frequency mode (LFM) VID is sent to the voltage regulator 212." *Id.*, ¶0078. This signal commands the "voltage regulator 212 to drive the core operational voltage back up to the required VID." *Id.*

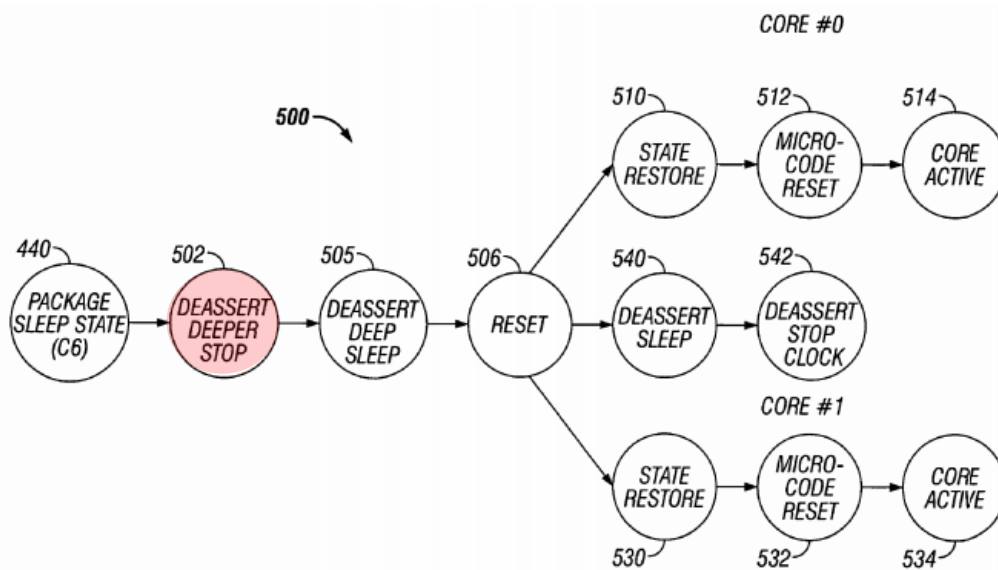


FIG. 5

EX1005, FIG. 5 (annotated).

Thus, Jahagirdar teaches or suggests this limitation. EX1003, ¶¶189-191.

m) Limitation [1.L]

As explained in Section VIII.A.2.i, upon receiving the low frequency mode (LFM) VID (i.e., power supply control signal), the voltage regulator powers the CPU by driving the core operational voltage back up to the required VID. See EX1005, ¶0078. Therefore, Jahagirdar discloses claim element [1.L]. EX1003, ¶¶192-193.

n) Limitation [1.M]

Jahagirdar teaches outputting a state restore signal (i.e., a restoration request signal) to the CPU and de-asserting a deep sleep signal (i.e., a clock supply control signal) which powers and engages the clocks.

In Sections VIII.A.2.d and VIII.A.2.g, Petitioner explained that the DPRSTP# signal teaches or suggests a power shutdown control signal. Here, the reverse is true. See EX1005, ¶0077 (describing an exit sequence that “may be considered the reverse of what happens during the previously-described entry phase.”). That is, when Jahagirdar’s control hub *de-asserts* the DPRSTP# signal, it acts as the restoration request signal. See EX1005, ¶0079 (“CPU 205 initiates an internal RESET” after the deep sleep (DPSLP#) signal is de-asserted and the PLLs are initiated), ¶0080 (“a state restore . . . is begun in preparation of the C6 exit as soon as the power and clocks are available”); EX1003, ¶¶194-195.

For example, at circle 502 of Figure 5 below, Jahagirdar's control hub deasserts the deeper stop signal (DPRSTP#), and a VID signal is sent to the voltage regulator, to command the "voltage regulator 212 to drive the core operational voltage back up to the required VID." EX1005, ¶0078.

Then "at a predetermined time, (e.g., controlled by timers)" Jahagirdar generates a clock supply control signal to assert the clocks back on. *Id.*, ¶0079. Jahagirdar also teaches an analog-to-digital converter (ADC) "as part of the zero voltage entry/exit logic 350 to monitor a voltage supply level and provide an associated digital output." *Id.*, ¶0032. Jahagirdar's use of the timer and the ADC ensures that the power supply voltage is stabilized before the clocks are turned on. EX1003, ¶¶196-199.

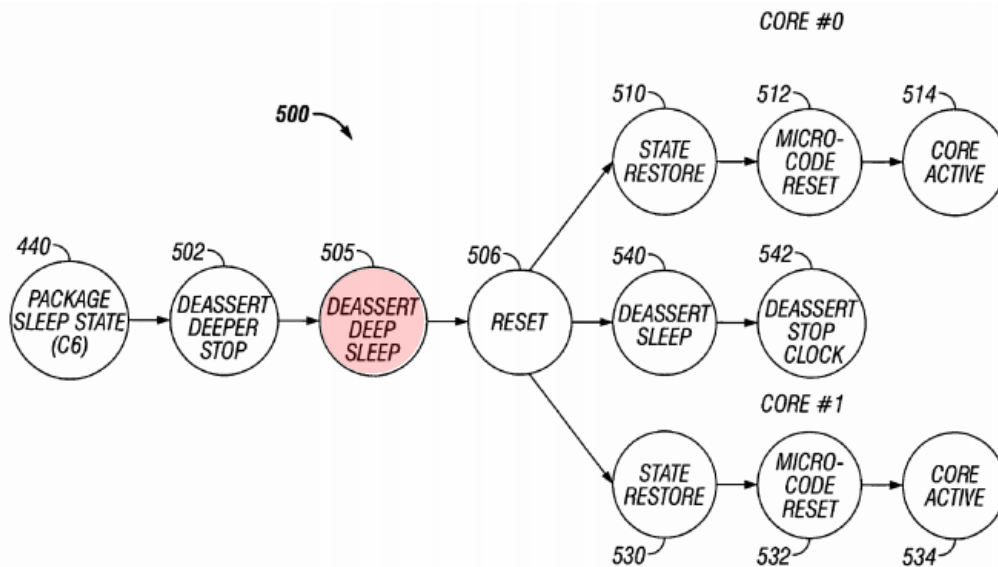


FIG. 5

EX1005, FIG. 5 (annotated).

In addition to the use of a timer and ADC that monitors the voltage supply and provides an associated output, because there is a clear delineation between the time when power is restored and when the clocks are asserted back on, it would have been obvious to wait for the power supply voltage to stabilize before advancing from circle 502 to 505 in Figure 5. EX1003, ¶201. And a POSITA would expect the clock restoration signal to be output before the CPU restoration signal because the CPU must wait until the clocks are restored. *Id.*

Therefore, Jahagirdar teaches or suggests this limitation. EX1003, ¶¶194-199.

o) Limitation [1.N]

As explained in Section VIII.A.2.n, at a “predetermined time (e.g. controlled by timers in the [control hub] 225), a signal to assert the clocks back on is asserted” and the system is reset. EX1005, ¶0079. After a reset is complete, “CPU 205 has the power and clocks engaged....” *Id.*, ¶¶0079, 0081. Therefore, Jahagirdar discloses claim element [1.N]. EX1003, ¶¶200-201.

p) Limitation [1.O]

As illustrated in Figure 5 below, after the power and clocks are available, and after “the states of the cores 320 and 322 are restored, CPU 205 is ready to continue from where it stopped.” EX1005, ¶¶0079-0080; *see supra* Sections VIII.A.2.l-o.

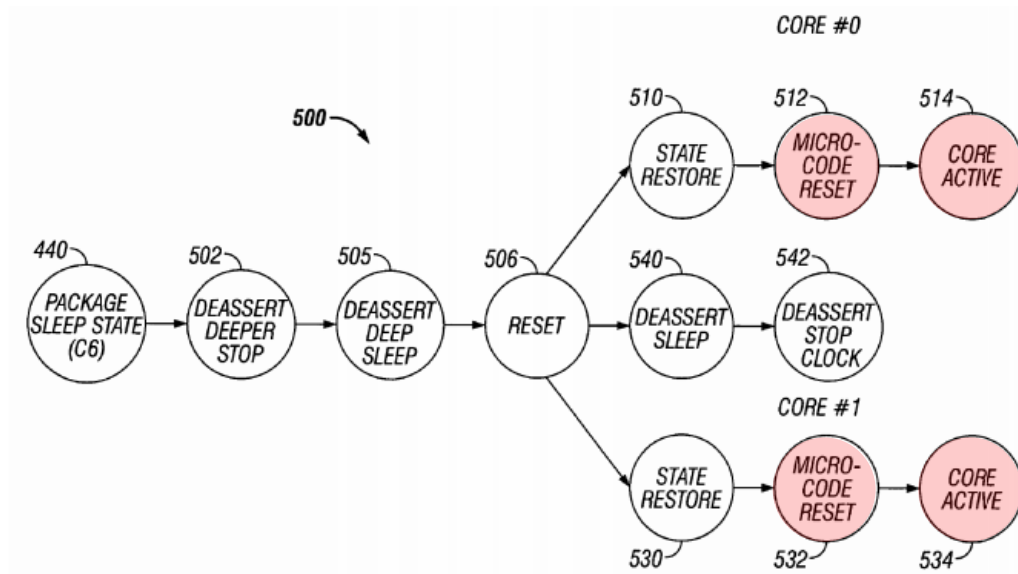


FIG. 5

EX1005, FIG. 5 (annotated).

Jahagirdar teaches that once “the states of the cores 320 and 322 are restored, CPU 205 is ready to continue from where it stopped.” EX1005, ¶[0080]. After Jahagirdar’s chipsets are powered on and the clocks are engaged, the CPU begins normal operation:

once the stop clock is de-asserted and the sleep signal is deasserted, both the CPU 205 and cores 320 and 322 and the chipset are all powered on and are operational, and both cores become active (circle 514 and circle 534). CPU 205 then typically begins some code fetches since its caches are empty and then will begin normal operations.

EX1005, ¶[0081].

ACPI also discloses a process for restoring information necessary in proceeding with the program. For example, ACPI teaches that “[c]ontrol starts

from the processor's reset vector after the wake event." EX1006, 0036. ACPI further states that "[o]n PCs, the wake function address is in memory below 1 MB and the control is transferred while in real mode. OSPM's wake function restores the processors' context." *Id.*, 0121. Additional examples of restoring processor context can be found throughout ACPI. *See id.*, 0222 ("The processor-complex context is maintained and...Dynamic RAM context is maintained and is read/write by the processors); 0243 ("OSPM will save context of all memory... Upon waking, OSPM will then restore the system context"); 0248 ("If this is an S2 or S3 wake, then the BIOS restores minimum context of the system before jumping to the waking vector."). EX1003, ¶¶202-204.

In sum, when the stop clock is de-asserted and the sleep signal is deasserted, the CPU 205 is powered on and is operational. Because the CPU architectural state (including a program counter) was saved, Jahagirdar's power management scheme allows the CPU to "continue[] where it left off in the execution stream" before the sleep mode was initiated. *See* EX1005, ¶¶0056, 0080-0081; *see also* EX1017, 201, explaining that a program counter is part of the state information necessary for continuing program execution after a program is suspended and restored.).

Thus, it would have been obvious to a POSITA that a program counter points to an address for processing to resume, and is an essential element of the architectural state of the machine. EX1003, ¶¶205-207. That is, a POSITA would

have understood that the most obvious way for Jahagirdar to have operated would have been to use a program counter to provide an address at which the CPU would “continue where it left off in the execution stream,” especially given that Jahagirdar expressly states that the architectural state is saved. EX1005, [0020], [0079]-[0080]; EX1003, ¶¶205-207. Therefore, it would have been obvious to branch a process to an address indicated by a program counter. EX1003, ¶¶205-207.

Thus, Jahagirdar teaches or suggests claim elements [1.O]. EX1003, ¶¶202-207.

3. Claim 2: The combination of Jahagirdar and ACPI renders this claim obvious.

As described in Section VIII.A.2.d, Jahagirdar in view of ACPI teaches or suggests that upon receiving the clock stop control signal from the power supply control unit, the power supply to the operation clock may be stopped. See EX1005, ¶0047 (“STOP_CPU signal may be asserted by the [control hub 225] and received by the clock generator 211 to cause the clock generator to halt the clock signal CLK to the CPU 205.”).

Jahagirdar in view of ACPI also teaches that upon receiving the clock stop control signal (STPCLK#) from the power supply control unit, the stop operation may be performed without stopping the supply of the operation clock. See *supra* VIII.A.2.d; EX1005, ¶0073; EX1003, ¶209. ACPI explains that the STPCLK#

signal may be used to implement “eight levels of clock control” to emulate different frequency settings, as follows:

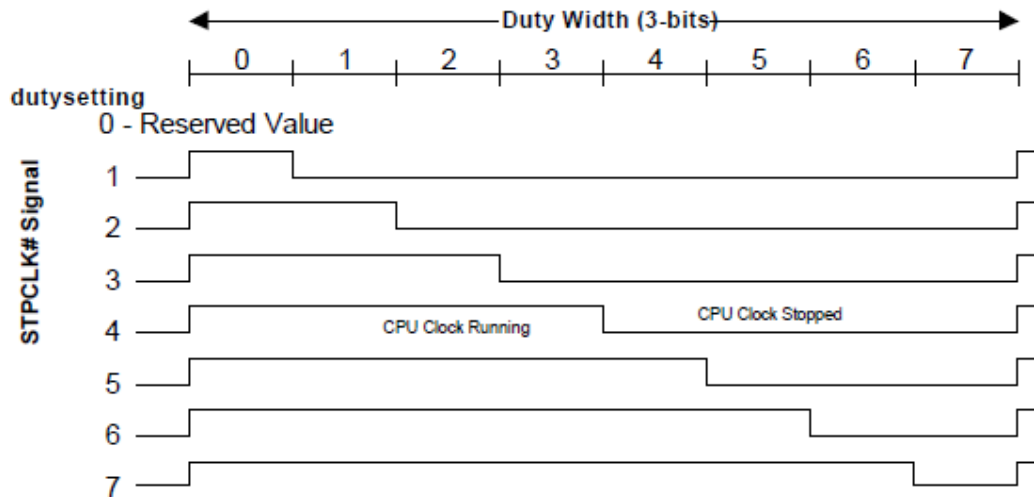


Figure 8-3 Example Control for the STPCLK#

EX1006, Figure 8-3, 0228.

Thus, the combination of Jahagirdar and ACPI teaches or suggests both alternatives recited in this claim. EX1003, ¶¶208-210.

4. Claim 3: The combination of Jahagirdar and ACPI renders this claim obvious.

The combination of Jahagirdar’s I/O controller hub and ACPI’s logic teaches or suggests the recited “power supply control unit.” Section VII.A.2.d; EX1003, ¶211.

As illustrated in Figure 4, Jahagirdar teaches that “each core 320 and 322 may execute an MWAIT instruction, and the initializing CPU core goes into a waiting state (e.g. CC6) and waits for the other core to get into the CC6 state as well, before the whole package (e.g. including both cores 320 and 322) may

transition into what may be termed the package C6 sleep state.” See EX1005, ¶¶0067-0069, 0071. Therefore, Jahagirdar’s waiting period teaches or suggests limiting “a period in which shutdown is possible,” as recited in claim 3.¹⁷ EX1003, ¶212.

In an additional example, Jahagirdar teaches limiting a time period in which shutdown is possible. See EX1005, ¶0008 (“ACPI may detect a time slot in which there are no new or pending interrupts”); EX1003, ¶¶213.

Accordingly, Jahagirdar teaches or suggests claim 3. EX1003, ¶¶211-214.

5. Claim 5: The combination of Jahagirdar and ACPI renders this claim obvious.

As explained in Section VIII.A.2.I, Jahagirdar teaches an exit sequence from the zero voltage processor sleep state that typically occurs when a break event or interrupt from the operating system or another source (i.e., a power supply restoration factor) necessitates waking up the CPU. EX1005, ¶¶0009, 0077. Jahagirdar’s ability to receive and respond to a break event or interrupt teaches or suggests the presence of an interrupt system. EX1003, ¶215. Because Jahagirdar expressly discloses waking the CPU in response to an interrupt, it would have been

¹⁷ This mapping is consistent with Patent Owner’s Infringement Contentions where Patent Owner asserts that waiting for a period of time before flushing CPU caches reads on this claim. EX1015, 0031, 0112-0113.

obvious that Jahagirdar includes an interrupt system, and that such systems are typically part of the I/O subsystem, such as the I/O Control Hub 225. *Id.*

Therefore, Jahagirdar teaches or suggests claim 5. EX1003, ¶¶215-217.

6. Claim 6: The combination of Jahagirdar and ACPI renders this claim obvious.

Jahagirdar and ACPI teaches the occurrence of a power shutdown factor. See *supra* VIII.A.2.g; see also EX1005, ¶¶0035, 0045-0052. For example, Jahagirdar discloses an event (i.e., power shutdown factor) where the CPU “may be transitioned to a lower power state...in response to firmware, such as microcode, or software, such as the operating system 245, or even ACPI software in some cases.” EX1005, ¶0045, ¶0009 (“Once the processor is placed into the deeper sleep state, a break event or interrupt from the operating system or another source may be sent to the chipset, and the chipset will then allow the processor to exit the deeper sleep state.”).

Jahagirdar may transition to a lower, or zero, power state in response to external power-management factors necessary for performance, power consumption, battery life, thermal requirements, and noise-level requirements. EX1005, ¶0045 (transitioning to “conserve power and/or reduce thermal load”); see also EX1006, §§ 8 (“Processor Control”), 12 (“Thermal Management”). A POSITA would have understood that external power-management signals would

be received at an input pin (i.e., an external terminal) of Jahagirdar's system.

EX1003, ¶¶218-219.

Furthermore, because Jahagirdar's microcode is stored on the CPU, a POSITA would have understood that the power shutdown events, triggered by Jahagirdar's microcode, teaches the recited state "provided by a command." EX1005, ¶¶0045, 0058; EX1003, ¶220. Because the Jahagirdar/ACPI system is configured to either act in response to an external shutdown request from an external terminal or by command, Jahagirdar/ACPI thus discloses a system configuration corresponding to both of the states. *Id.*

Therefore, Jahagirdar teaches or suggests this claim. EX1003, ¶¶218-221.

7. Claim 7: The combination of Jahagirdar and ACPI renders this claim obvious.

As explained in Section VIII.A.2.p, Jahagirdar teaches a microprogram for restoring the evacuated information necessary (i.e., limitation [7.B]) for the CPU to "continue where it left off in the execution stream" before the sleep mode was initiated (i.e., proceed with the program). *See* EX1005, ¶¶0054, 0056, 0080-0081; EX1003, ¶222.

With respect to limitation [7.A], ACPI discloses a microprogram for power management in a microcontroller. ACPI teaches several sleeping states, S2-S5 and details the process of transiting to and from each sleep state. *See* EX1006, 0036,

0239-0245. ACPI refers to “restoration” as transitioning (or waking or reset) from a sleep state to a working state. *Id.*, 0239-0240; EX1003, ¶223.

The ACPI C-states correspond to CPU power states while the entire system is in a global “working state” (G0 state), whereas the S-states refer to system sleep states while the entire system, including the CPU, is in a global “sleep state” (G1 state). EX1006, 0063, 0073; EX1003, ¶229. However, like the C-states, ACPI’s S-states control both CPU power and CPU wake characteristics. For example in the S1, “the processors are not executing instructions,” “[t]he processor-complex context is maintained,” the “dynamic RAM context is maintained,” and the transition from S1 state to wake “causes the processor to continue execution where it left off.” EX1006, 0222. A POSITA would have understood that the control asserted over the CPU in the S1 state resembles the control asserted in the C-states. EX1003, ¶229.

With respect to limitation [7.B], when transitioning from any of sleep states S2-S4, ACPI teaches that the evacuated information is restored. *See* EX1006, 0243 (“platform context is maintained” and “[u]pon waking, OSPM will then restore the system context.”). ACPI further explains “When the system wakes, the firmware performs a normal boot process and transfers control to the OS via the firmware-waking-vector loader. The OS then restores the system’s memory and resumes execution.” EX1006, 0223; EX1003, ¶224.

ACPI's power-on reset operation teaches or suggests limitations [7.C] and [7.D]. EX1003, ¶225. When transitioning from sleep state S5 (also called a "soft off state"), ACPI teaches that upon power on, a normal power-on reset operation is performed based on a restored set value¹⁸, and the evacuated information is not restored. See EX1006, 0244. ACPI's processor "will start executing at its power-on reset vector when waking from an S2, S3, or S4 sleeping state, during a power-on sequence, or as a result of a hard or soft reset." *Id.*, 0248. When executing from the power-on reset vector as a result of a power-on sequence, a hard or soft reset, or waking from an S4 sleep state, the platform firmware performs complete hardware initialization; placing the system in a boot configuration. See *id.*, 0246-0249. The firmware then passes control to the operating system boot loader. See *id.*

¹⁸ The '575 patent fails to provide a definition for a "set value," however Patent Owner argues that "a standard reset" is both a "a microprogram for restoring a set value without using the evacuated information," and "a program from resetting processes." EX1015, 0033-0035.

Thus, ACPI's power-on sequence, hard reset, or soft reset discloses both a program for restoring a set value without using the evacuated information and a resetting processes.¹⁹ See EX1006, 0244, 0248; EX1003, ¶¶225-226.

ACPI also provides a program for resetting processes, using a set value, via a reset register. EX1006, 0091 ("The optional ACPI reset mechanism specifies a standard mechanism that provides a complete system reset. . . . To reset the machine, software will write a value (indicated in RESET_VALUE in FADT) to the reset register."); EX1003, ¶227.

With respect to limitation [7.E], ACPI discloses different options for how the CPU may operate:

This section covers the initialization sequences for an ACPI platform. After a reset or wake from an S2, S3, or S4 sleeping state (as defined by the ACPI sleeping state definitions), the CPU will start execution from its boot vector. At this point, the initialization software has many options, depending on what the hardware platform supports. This section describes at a high level what should be done for these different options. Figure 9-2 illustrates the flow of the boot-up software.

¹⁹ This mapping is consistent with Patent Owner's Infringement Contentions where Patent Owner argues that "a standard reset" meets both [7.C] and [7.D]. EX1015, 0033-0035.

EX1006, 0246.

Furthermore, when the “platform initializes from a cold boot (mechanical off or from an S4 or S5 state), the hardware platform may be configured in a legacy configuration,” and will enable ACPI mode. *Id.*, 0248. Thus, the combination of Jahagirdar and ACPI teaches that Jahagirdar’s CPU selects one of the plurality of microprograms in an alternative way, based on its current sleep state and the type of wake signal received. Therefore, the programs are determined based on “a power shutdown factor, a register, signal, terminal, or the like.” *See supra* VII.D.9. Thus, the combination of Jahagirdar and ACPI teaches or suggests each feature of claim 7. EX1003, ¶¶228-229, 231.

It would have been obvious for a POSITA to enhance the C-states of Jahagirdar and ACPI, with the power-on reset of ACPI’s S-states. EX1003, ¶230. A POSITA would have understood that having a reboot feature, like that of ACPI’s S-states, offers many advantages. EX1003, ¶230 (citing EX1019, Candea, 1, (explaining that rebooting “has been around as long as computers themselves” and offers advantages such as resolving a “deadlock,” working around “Heisenbugs,” reclaiming “stale resources,” cleaning up corrupted states, etc.)). A POSITA would have also understood that the power-on reset controls asserted over the CPU in the S-states could easily be implemented and would be desirable to be implemented in the context of the C-states. EX1003, ¶230.

8. Claim 11: The combination of Jahagirdar and ACPI renders this claim obvious.

Jahagirdar teaches that “voltage from the I/O control hub 225 ($V_{I/O}$ 349) may be provided to the processor 205 in order to provide sufficient power to the dedicated cache memory 340 such that it can store the critical state variables associated with the processor 205 while the rest of the processor 205 is powered down...” EX1005, ¶0036, emphasis added; *see also id.*, ¶0007 (describing a “lower data retention point” to conserve energy); EX1003, ¶232.

Jahagirdar’s I/O control hub voltage ($V_{I/O}$ 349) is effectively a second power supply unit that provides sufficient power to a circuit portion outside a power shutdown range for information holding when the power shutdown factor occurs, because the I/O control hub voltage ($V_{I/O}$ 349) maintains power while the rest of the processor is powered down. EX1003, ¶233; Section VIII.A.2.g.

Therefore Jahagirdar renders this claim obvious. EX1003, ¶¶232-234.

9. Claim 15: The combination of Jahagirdar and ACPI renders this claim obvious.

As explained in Section VIII.A.2.c, Jahagirdar discloses a voltage regulator (i.e., power supply unit) arranged between the CPU and an external battery 255 (i.e., power supply device) for supplying an operating voltage to the CPU. *See* EX1005, ¶0033. Thus, Jahagirdar teaches this claim. EX1003, ¶¶235-236.

10. Claim 16: The combination of Jahagirdar and ACPI renders this claim obvious.

As recited in claim 1, the clock stop control signal occurs after the power shutdown control signal from the power supply control unit has been received. Accordingly, it is understood that the power supply shutdown control signal is not intended to shut down the power supply to the clock generator. EX1003, ¶237.

ACPI's clock generator may be configured to select an operation stop performed without stopping the supply of the operation clock upon receiving the clock stop control signal from the power supply control unit. See Section VII.A.3; EX1005, ¶0073; EX1006, 0228.

Furthermore, Jahagirdar's "power supply unit" includes both the core VCC and clock generator voltage supplies. EX1003, ¶¶238-239. A POSITA would have understood the clock generator to be powered by a secondary voltage supply, such as an input/output voltage ($V_{DDI/O}$ or $V_{I/O}$) or an auxiliary voltage (V_{DDA}).²⁰ EX1003, ¶239 (citing EX1025, Intel Platform Design Guide, 52-53, 204).

Thus, because the clock generator maintains power upon receiving upon receiving the power shutdown control signal, as recited in claim 1, and because

²⁰ This mapping is consistent with Patent Owner's Infringement Contentions where Patent Owner has mapped the clock generator to include a voltage regulator (core/PLL voltage regulator). EX1015, 0010-0011.

Jahagirdar's clock generator may implement various frequency settings upon receiving a clock stop signal, the combination of Jahagirdar and ACPI teaches or suggests this claim. EX1003, ¶¶237-240.

B. Ground 2: Claims 4, 17-22, 26, 30, and 31 are obvious over Jahagirdar, ACPI, and Saito²¹.

Saito discloses a “power supply cut recovery system in a data processing device wherein, when a power supply OFF occurs, the state of progress of a program during execution is stored and, upon restoration of the power supply, execution of the program is continued, comprising... a plurality of stack areas for storing a program state of progress when the power supply is OFF...” EX1010, Saito, 0001.

²¹ Saito appears on the face of the '575 patent, but it appears that the Examiner only reviewed a translated abstract. EX1001, (56); EX1002, 12-18. The Examiner did not substantively consider or address Saito during prosecution. EX1002, 12-18. Therefore, the same or substantially the same prior art or arguments were *not* previously presented to the Office and the Board should *not* invoke its discretionary power under 35 U.S.C. §325(d). *See Becton, Dickinson and Co. v. B. Braun Melsungen AG*, IPR2017-01586, Paper 8, 17-18 (P.T.A.B. Dec. 15, 2017)(informative).

1. Motivation to combine Jahagirdar, ACPI, and Saito.

As explained in Section VIII.A.1, a POSITA would have been motivated to combine Jahagirdar and ACPI, because Jahagirdar expressly uses and implements power-management concepts disclosed in ACPI. EX1003, ¶241. A POSITA would have been further motivated to combine Jahagirdar/ACPI and Saito, because each are in the same field of controlling power management of a CPU, and each attempts to solve the age-old problem of reducing a system's power consumption while maintaining accurate state information, as does the challenged '575 Patent. EX1005, Abstract, ¶0003; EX1010, 0001; EX1003, ¶242. In addition, Jahagirdar/ACPI discloses techniques for evacuating the information necessary in proceeding with the program to a dedicated cache memory and storing an address so the CPU can continue program execution from a shutdown state and resume where it left off. *See supra* Sections VIII.A.2.e, VIII.A.2.p.

As illustrated in Jahagirdar's Figure 3, Jahagirdar/ACPI discloses a base register that, under microcode control, indexes read/write access to register 384. EX1005, ¶¶0063-0064. That is, base register 382 may be used to increment register 384 which operates on SRAM 340 by a 12 bit pointer and a 2 bit read/write enable in order to store the evacuated data. *Id.*, ¶0064. For this reason, Jahagirdar's register 384 does not support the decrementing function typical of a last-in, first-out (LIFO) or stack structure. Implementing a queue versus stack structure is

simply a matter of design choice, and Jahagirdar could be easily modified to implement the latter. EX1003, ¶243.

A POSITA looking to implement a stack would have looked to Saito since it is in the same field of power management. EX1003, ¶243 (citing EX1008, Crump, 22:36-41, 37:11-33, 37:47-48); *see* EX1010, 0001-0002. And a POSITA would have understood the decrementing teachings of Saito to merely fill in obvious details not disclosed in Jahagirdar/ACPI. EX1003, ¶243.

Specifically, Saito teaches a stack pointer to point to a next address of a stack region. EX1010, 0001-0002. Because stacks are usually maintained as a last in, first out structure, as data is stored on Saito's stack, a POSITA would have understood that its stack pointer is decremented before an item is placed on the stack. *See id.*, 2-3; EX1003, ¶244. Upon restore, when Saito's decremented stack pointer points to a power supply cut program (i.e., the program run, that caused the shutdown), "the stack pointer level is increased by 1, to read out the next address (1) that was stored in the previous cycle. EX1010, 0002-0003. It is possible to read out the address for the processing program that is to be run and it "will be possible to continue processing from the state prior to the power supply OFF." *Id.*, 0003; EX1003, ¶244.

Indeed, the '575 patent confirms that evacuating information to an information holding circuit were previously used in power management systems.

EX1001, 1:40-55. For these additional reasons, a POSITA would have been motivated to combine Jahagirdar/ACPI and Saito. EX1003, ¶245.

Moreover, because the power management system of Jahagirdar/ACPI and Saito are designed to operate in similar manners (e.g., evacuate the necessary information to a dedicated cache memory). EX1005, ¶¶0054, 0062; EX1010, 0002 (“save region of the RAM 8”). A POSITA could have enhanced Jahagirdar’s register and expanded its functionality to include incrementing, resetting, and decrementing using the stack register and stack pointer of Saito with the dedicated cache memory of Jahagirdar/ACPI using known methods. EX1003, ¶246. The substitution would have been routine, and would have led to the predictable result of storing the evacuated information to a stack region of the dedicated cache memory with a stack pointer, that could be decremented in addition to incremented. EX1003, ¶246.

Finally, the proposed combination of Jahagirdar/ACPI and Saito would involve nothing more than incorporating a well-known device (i.e., Saito’s stack) into a well-known system (i.e., Jahagirdar’s cache memory). EX1003, ¶247; *KSR* 550 U.S. at 419-421 (“[I]n many cases a person of ordinary skill [in the art] will be able to fit the teachings of multiple patents together like pieces of a puzzle.”).

In sum, a POSITA would have been motivated to make the minor modification required to Jahagirdar/ACPI to include Saito’s stack register.

EX1003, ¶¶244-250. The resulting combination is little more than use of a known technique to improve power consumption while maintaining necessary data. And after the combination, each feature would continue to function as intended. *See* KSR 550 U.S. at 417. EX1003, ¶248.

2. Claim 4: The combination of Jahagirdar, ACPI and Saito renders this claim obvious.

Jahagirdar teaches evacuating the information necessary in proceeding with the program to a dedicated cache memory and stores an address so the CPU can continue program execution from a shutdown state and resume where it left off. *See supra* Sections VIII.A.2.e, VIII.A.2.p.

Saito discloses that “the state of progress of a program during execution is stored and, upon restoration of the power supply, execution of the program is continued.” EX1010, 0002. This is achieved using “a plurality of stack areas for storing a program state of progress when the power supply is OFF.” *Id.* When power is restored, the first program counter address that is indicated by the stack pointer SP will be the power supply cut program that caused the processor to turn off. *Id.* Saito’s “stack pointer level is increased by 1, to read out the next address (1) that was stored in the previous cycle. EX1010, 0002; EX1003, ¶¶249-251.

In this way Saito evacuates the information necessary in proceeding with the program to a stack region in the information holding unit and then evacuates to a specific place only a value of a stack pointer for pointing to an evacuation place.

EX1003, ¶253. Thus, the combination of Jahagirdar, ACPI, and Saito teaches this claim. EX1003, ¶¶249-253.

3. Claim 17: The combination of Jahagirdar, ACPI and Saito renders this claim obvious.

a) Preamble and limitations [17.A]-[17.F]

Limitations [17.Preamble]-[17.F] are identical to limitations [1.Preamble]-[1.D],[1.F]-[1.G] above. Thus, the Jahagirdar and ACPI combination teaches or suggests these limitations for at least the reasons described above. See Sections VIII.A.2.a-e,VIII.A.2.g-h; EX1003, ¶¶254-255.

b) Limitation [17.G]

As explained above, Jahagirdar outputs an evacuation complete signal, Section VIII.A.2.h, and it receives an occurrence of a power supply restoration factor. Section VIII.A.2.i. A POSITA would have understood that Jahagirdar's evacuation complete signal would have been received before receiving an occurrence of the power supply restoration factor. EX1003, ¶256.

Further, as explained in Section VIII.A.2.n, Jahagirdar teaches outputting a state restore signal (i.e., a restoration request signal) to the CPU. Thus, at least for the reasons described above, Jahagirdar teaches or suggests this limitation. EX1003, ¶257.

c) Limitation [17.H]

As explained in Sections VIII.A.2.p and VIII.A.7, Jahagirdar discloses a power supply restoration microprogram that is executed upon receiving a restoration request signal.

Jahagirdar further discloses that the “state of CPU core 320 is saved at circle 406 to dedicated cache memory.” EX1005, ¶0068. Yet, the state of the CPU is maintained as the CPU transitions between states C1 and at least C4. *Id.*, ¶0007 (stating “C4 is defined as a Deeper Sleep state in which all processor clocks are stopped and the processor voltage is reduced to a lower data retention point.”); ¶0077 (Jahagirdar’s exit sequence “may be considered the reverse of what happens during the previously-described entry phase.”); EX1003, ¶259. ACPI also explains that when in states C1-C3, the “hardware can exit this state for any reason, but must always exit this state whenever an interrupt is to be presented to the processor.” EX1006, 0229.

Thus, if an interrupt occurred during any of states C1-C3, the evacuated information necessary in proceeding with the program would still be stored at the CPU and therefore would not need to be restored. EX1005, ¶0007; EX1006, 0229; EX1003, ¶260. Because the evacuated information need not be restored, the CPU could resume execution without restoring the evacuated information. EX1003, ¶260.

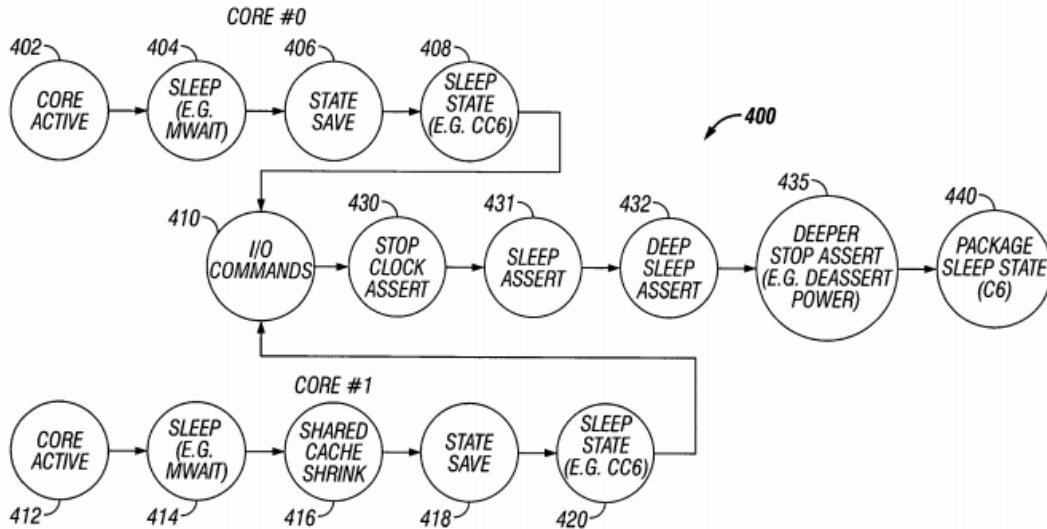


FIG. 4

EX1005, FIG. 4.

Furthermore, as explained in Section VIII.A.7, ACPI teaches a microprogram that does not restore the evacuated information. EX1003, ¶261.

Thus, a POSITA would have understood that the combination of Jahagirdar and ACPI teaches or suggests this limitation. EX1003, ¶¶258-262.

d)Limitation [17.I]

Limitation [17.I] is substantially identical to claim 4. EX1003, ¶263. Thus, the Jahagirdar, ACPI, and Saito combination teaches or suggests this limitation for at least the reasons described above. See Section VII.B.2; EX1003, ¶263.

4. Claims 18-22, 26, 30, and 31 are obvious in view of Jahagirdar, ACPI, and Saito.

The combination of Jahagirdar, ACPI, and Saito teaches or suggests claim 17. See Section VIII.B.3.

Furthermore, claims 18-22, 26, 30, and 31 are substantially identical to claims 2-3, 5-7, 11, 15, and 16, respectively. EX1003, ¶265. The difference between the claims being only their dependencies on different independent claims. *Id.* Thus, the combination of Jahagirdar, ACPI, and Saito teaches or suggests these claims for at least the reasons described above. See Sections VIII.A.3-10; EX1003, ¶¶264-265.

As explained in Section VIII.B.1, a POSITA would have been motivated to combine Jahagirdar, ACPI, and Saito. EX1003, ¶266.

C. Ground 3: Claim 12 is obvious over Jahagirdar, ACPI, and Chung.

1. Motivation to Combine Jahagirdar, ACPI, and Chung

As described in Section VIII.A.1, a POSITA would have been motivated to combine Jahagirdar and ACPI, because Jahagirdar expressly uses and implements power-management concepts disclosed in ACPI. EX1003, ¶140-143, 267. A POSITA would have been further motivated to combine Jahagirdar/ACPI and Chung, because each are in the same field of controlling power management of a CPU, and each attempts to reduce a system's power consumption while maintaining accurate state information. EX1005, Abstract, ¶0003; EX1010, 0001; EX1007, Abstract, 1:9-12; EX1003, ¶267.

As described throughout Sections VIII.A.2.d and VIII.A.2.h-i, Jahagirdar's I/O control hub outputs a power shutdown control signal and its clock/power

management unit 350 responds by switching the CPU VID signal to a zero voltage level. EX1005, ¶0074; *see also id.*, ¶0048-0050 (disclosing that one or more power management-related signals causes clock/power management logic 350 to initiate a zero voltage level). However, Jahagirdar does not disclose a circuit for loading an inverted value to the clock/power management logic registers. EX1003, ¶268

A POSITA would have understood that circuits for inverting signals are common and that employing an inverted circuit, like that of Chung, increases the number of input state signal possibilities. EX1003, ¶269. A POSITA could have enhanced Jahagirdar's clock/power management logic registers and expanded the functionality to include loading opposite Boolean values, as described in Chung. EX1003, ¶269. The substitution would have been routine, and would have led to the predictable result of sending/loading an inverted signal. EX1003, ¶269.

Finally, the proposed combination of Jahagirdar/ACPI and Chung would involve nothing more than incorporating a well-known device (i.e., Chung's inverter circuit) into a well-known system (i.e., Jahagirdar's clock/power management logic). EX1003, ¶270; KSR 550 U.S. at 419-421 (“[I]n many cases a person of ordinary skill [in the art] will be able to fit the teachings of multiple patents together like pieces of a puzzle.”).

In sum, a POSITA would have been motivated to make the minor modification required to Jahagirdar/ACPI to include Chung's inverter circuit.

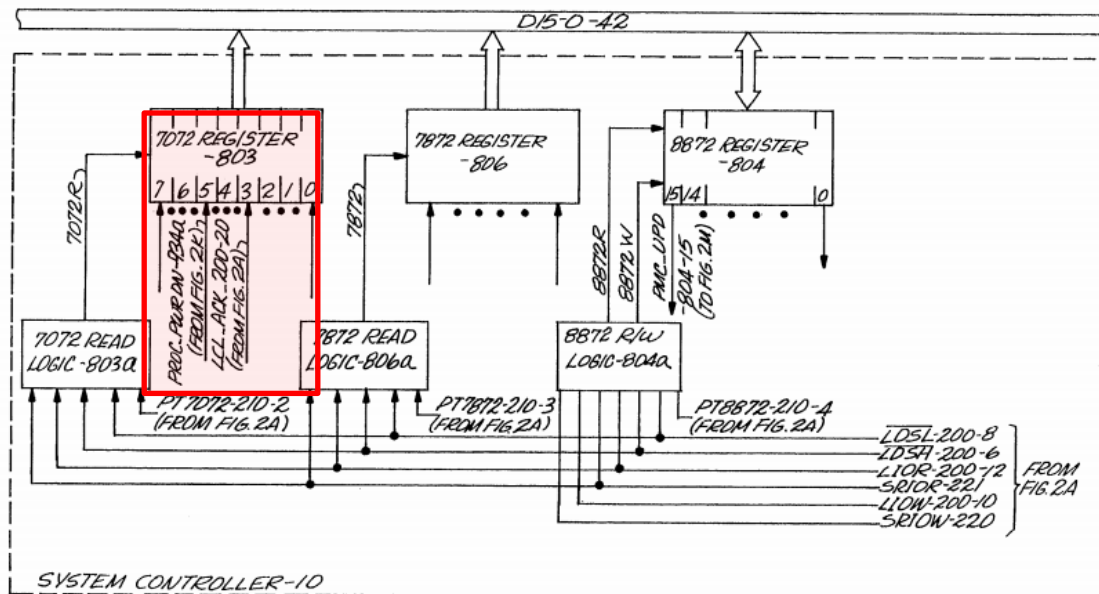
EX1003, ¶¶267-271. The resulting combination is little more than use of a known technique to improve power consumption while maintaining necessary data. And after the combination, each feature would continue to function as intended. *See* KSR 550 U.S. at 417. EX1003, ¶271.

2. Claim 12

Chung teaches writing a value into PMCOUT latches 74 (i.e., a power shutdown control signal) to thereby command the CPU power supply to be disabled at power switch 1002. EX1007, 32:14-19. Because the foregoing circuits directly control the power supplied to Chung's CPU, they are in the power shutdown range. EX1003, ¶272.

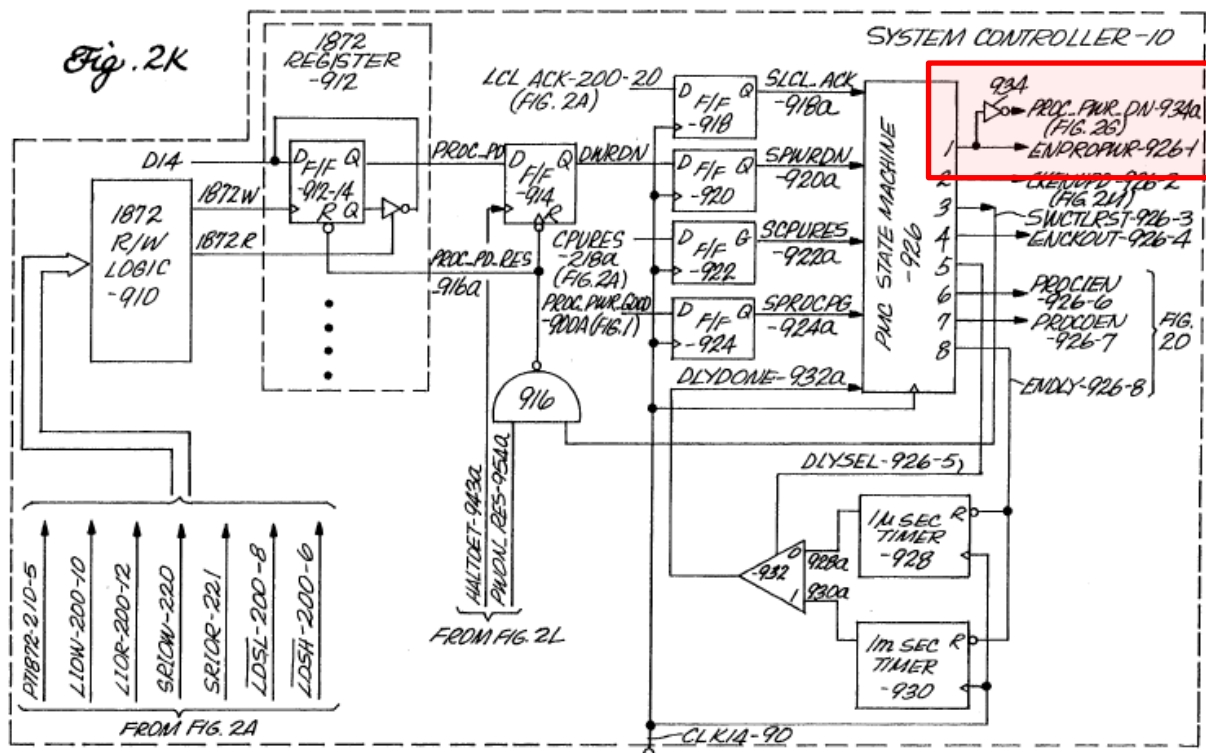
As illustrated in Figures 2G and 2K below, Chung teaches that PMCOUT latches 74 obtains its data from latch 5 of the 7072 register 803 that holds the PROC-PWR-DN 934a signal, which is also a circuit portion in a power shutdown range. Ex1007, , 33:16-23, FIG. 2G; *see also id.*, 7:67-8:4 (“[p]ower management control timing and logic interfaces with external multiplexers and latches, and has the ability to power down the main processor or the main processor and the input/output devices and controllers, conserving power essential to portable laptop computers.”). The PROC-PWR-DN 934a signal is formed by passing the ENPROPWR 926-1 through inverter 934. *Id.*, 33:16-23, FIG. 2K.

Fig. 26



EX1007, FIG. 2G (annotated).

Fig. 2K



EX1007, FIG. 2K (annotated).

Chung therefore teaches or suggests this claim by inverting the ENPROPWR 926-1 to form the PROC-PWR-DN 934a signal that, after being written to register 803, signals PMCOUT latches⁷⁴ to shut down power to the CPU. EX1007, 32:14-17, 33:16-23; EX1003, ¶¶272-274. Therefore the combination of Jahagirdar, ACPI, and Chung teaches this claim.

D. Ground 4: Claim 27 is obvious over Jahagirdar, ACPI, Saito and Chung.

As described in Section VII.A.1, Jahagirdar suggests the proposed combination with ACPI. A POSITA would have been motivated to combine Jahagirdar/ACPI and Saito and Jahagirdar/ACPI and Chung, for the reasons provided in Sections VII.B.1 and VII.C.1. A POSITA would have been further motivated to combine Jahagirdar/ACPI/Saito and Chung at least for the reasons described above. See Sections VII.A.1, VII.B.1, VII.C.1; EX1003, ¶275.

Claim 27 is substantially identical to claim 12, the difference between the claims being only their dependencies on different independent claims. EX1003, ¶276. Thus, the combination of Jahagirdar/ACPI/Saito and Chung teaches or suggests claim 27 for at least the reasons described above. See Section VII.D.2; EX1003, ¶276.

IX. Conclusion

This Petition demonstrates a reasonable likelihood that the challenged claims are unpatentable. Thus, Petitioner respectfully requests institution of *inter partes* review.

Respectfully submitted,

STERNE, KESSLER, GOLDSTEIN & FOX P.L.L.C.

/Michael B. Ray/

Date: September 16, 2019
1100 New York Avenue, N.W.
Washington, D.C. 20005-3934
(202) 371-2600

Michael B. Ray (Registration No. 33,997)
Trent W. Merrell (Registration No. 73,771)
Michael D. Specht (Registration No. 54,463)
Jonathan Tuminaro (Registration No. 61,327)
Attorneys for Petitioner

**CERTIFICATION OF COMPLIANCE WITH TYPE-VOLUME
LIMITATION, TYPEFACE REQUIREMENTS, AND TYPE STYLE
REQUIREMENTS (37 CFR § 42.24(d))**

1. This Petition complies with the type-volume limitation of 14,000 words, comprising 13,657 words, excluding the parts exempted by 37 C.F.R. § 42.24(a).

2. This Petition complies with the general format requirements of 37 C.F.R. § 42.6(a) and has been prepared using Microsoft® Word 2010 in 14 point Times New Roman.

STERNE, KESSLER, GOLDSTEIN & FOX P.L.L.C.

/Michael B. Ray/

Date: September 16, 2019
1100 New York Avenue, N.W.
Washington, D.C. 20005-3934
(202) 371-2600

Michael B. Ray (Registration No. 33,997)
Attorney for Petitioner

CERTIFICATION OF SERVICE (37 C.F.R. §§42.6(e), 42.105(a))

The undersigned hereby certifies that true and correct copies of the foregoing **PETITION FOR INTER PARTES REVIEW**, its accompanying Exhibits 1001-1026, and Petitioner's Power of Attorney were served via Express Mail on September 16, 2019, in their entirety on the following:

McDermott Will and Emery LLP
The McDermott Building
500 North Capitol Street, N.W.
Washington, DC 20001

FREITAS & WEINBERG LLP
350 Marine Parkway, Suite 200
Redwood Shores, CA 94065

*Additional Address Known to Petitioner
as Likely to Effect Service*

*Patent owner's correspondence
address of record for U.S. Patent No.
7,930,575*

The above-captioned **PETITION FOR INTER PARTES REVIEW**, its accompanying Exhibits 1001-1026, and Petitioner's Power of Attorney were also served via e-mail on September 16, 2019, in their entirety on the following:

Robert E. Freitas
Jing H. Cherng
FREITAS & WEINBERG LLP
rfreitas@fawlaw.com
gcherng@fawlaw.com

STERNE, KESSLER, GOLDSTEIN & FOX P.L.L.C.

Date: September 16, 2019
1100 New York Avenue, N.W.
Washington, D.C. 20005-3934
(202) 371-2600

/Michael B. Ray/

Michael B. Ray (Registration No. 33,997)
Attorney for Petitioner

Appendix A

Claim listing:

Claim:	Limitation:	Claim Language:
Claim 1	[1.Preamble]	A microcontroller comprising:
	[1.A]	a CPU;
	[1.B]	a power supply unit arranged between the CPU and a power supply device for supplying power to the CPU;
	[1.C]	a power supply control unit for controlling the CPU and the power supply unit;
	[1.D]	an information holding unit for holding information evacuated from the CPU, the information being necessary in proceeding with a program;
	[1.E]	a clock generator, wherein:
	[1.F]	the power supply control unit outputs a shutdown request signal to the CPU in response to an occurrence of a power shutdown factor,
	[1.Ga-c]	the CPU, upon receiving the shutdown request signal, [a] executes a power shutdown microprogram, [b] evacuates the information necessary in proceeding with the program to the information holding unit, and [c] outputs an evacuation completed signal to the power supply control unit after evacuation is completed,
	[1.H]	the power supply control unit, upon receiving the evacuation completed signal, outputs a power shutdown control signal to the power supply unit,
	[1.I]	the power supply unit shuts down power supply to the CPU upon receiving the power shutdown control signal from the power supply control unit,

Inter Partes Review of U.S. Patent No. 7,930,575 B2

	[1.J]	the power supply control unit, upon receiving the evacuation completed signal, outputs a clock stop control signal to the clock generator,
	[1.K]	the power supply control unit outputs a power supply control signal to the power supply unit in response to an occurrence of a power supply restoration factor,
	[1.L]	the power supply unit, upon receiving the power supply control signal, starts to supply power to the CPU,
	[1.M]	the power supply control unit outputs a restoration request signal to the CPU and outputs a clock supply control signal to the clock generator when a supplied power supply voltage is stabilized,
	[1.N]	the clock generator, upon receiving the clock supply control signal, restarts to supply an operation clock, and
	[1.O]	the CPU, upon receiving the restoration request signal, executes a power supply restoration microprogram, restores the information necessary in proceeding with the program and evacuated in the information holding unit in time of power shutdown, and then branches a process to an address indicated by a program counter to continue program execution from a shutdown state.
Claim 2	--	The microcontroller according to claim 1, wherein the clock generator selects one of [2.A] operation performed after stopping supply of an operation clock and [2.B] operation stop performed without stopping the supply of the operation clock [2.C] as operation to be performed upon receiving the clock stop control signal from the power supply control unit.
Claim 3	--	The microcontroller according to claim 1, wherein the power supply control unit limits a period in which shutdown is possible.

Inter Partes Review of U.S. Patent No. 7,930,575 B2

Claim 4	--	The microcontroller according to claim 1, wherein the power shutdown microprogram evacuates the information necessary in proceeding with the program to a stack region in the information holding unit and then evacuates to a specific place only a value of a stack pointer for pointing to an evacuation place.
Claim 5	--	The microcontroller according to claim 1, wherein the power supply restoration factor uses an interrupt system of the microcontroller.
Claim 6	--	The microcontroller according to claim 1, wherein a state of the power shutdown factor includes at least one of [6.A] a state where it is provided from an external terminal and [6.B] a state where it is provided by a command, and it is possible to [6.C] select a configuration corresponding to either one or both of the states.
Claim 7	--	The microcontroller according to claim 1, wherein: [7.A] the power supply restoration microprogram includes: [7.B] a microprogram for restoring the evacuated information necessary in proceeding with the program and proceeding with the program; [7.C] a microprogram for restoring a set value without using the evacuated information necessary in proceeding with the program; and [7.D] a program for resetting processes, and [7.E] the CPU selects one of the plurality of microprograms in an alternative way.
Claim 11	--	The microcontroller according to claim 1, further comprising a second power supply unit for lowering a voltage to be supplied to a circuit portion outside a power shutdown range to a lower limit voltage for information holding when the power shutdown factor occurs.
Claim 12		The microcontroller according to claim 1, further comprising a circuit for loading an inverted value to a register in a circuit portion in a power shutdown

Inter Partes Review of U.S. Patent No. 7,930,575 B2

		range.
Claim 15	--	The microcontroller of claim 1, wherein the power supply unit includes a voltage regulator.
Claim 16	--	The microcontroller of claim 1, wherein the power supply unit does not shut down the power supply at least to the clock generator upon receiving the power shutdown control signal from the power supply control unit.
Claim 17	[17.Preamble]	A microcontroller comprising:
	[17.A]	a CPU;
	[17.B]	a power supply unit arranged between the CPU and a power supply device for supplying power to the CPU;
	[17.C]	a power supply control unit for controlling the CPU and the power supply unit; and
	[17.D]	an information holding unit for holding information evacuated from the CPU, the information being necessary in proceeding with a program; wherein:
	[17.E]	the power supply control unit outputs a shutdown request signal to the CPU in response to an occurrence of a power shutdown factor,
	[17.F]	the CPU, upon receiving the shutdown request signal, executes a power shutdown microprogram, evacuates the information necessary in proceeding with the program to the information holding unit, and outputs an evacuation completed signal to the power supply control unit after evacuation is completed,
	[17.G]	the power supply control unit, upon receiving an occurrence of a power supply restoration factor after receiving the evacuation completed signal, outputs a restoration request signal to the CPU,
	[17.H]	the CPU, upon receiving the restoration request

Inter Partes Review of U.S. Patent No. 7,930,575 B2

		signal, executes a power supply restoration microprogram which does not restore the evacuated information, and continues program execution, and
	[17.I]	the power shutdown microprogram evacuates the information necessary in proceeding with the program to a stack region in the information holding unit, and then evacuates to a specific place only a value of a stack pointer for pointing to an evacuation place.
Claim 18	--	The microcontroller according to claim 17, wherein the clock generator selects one of operation performed after stopping supply of an operation clock and operation stop performed without stopping the supply of the operation clock as operation to be performed upon receiving the clock stop control signal from the power supply control unit.
Claim 19	--	The microcontroller according to claim 17, wherein the power supply control unit limits a period in which shutdown is possible.
Claim 20	--	The microcontroller according to claim 17, wherein the power supply restoration factor uses an interrupt system of the microcontroller.
Claim 21	--	The microcontroller according to claim 17, wherein a state of the power shutdown factor includes at least one of a state where it is provided from an external terminal and a state where it is provided by a command, and it is possible to select a configuration corresponding to either one or both of the states.
Claim 22	--	The microcontroller according to claim 17, wherein the power supply restoration microprogram includes: a microprogram for restoring the evacuated information necessary in proceeding with the program and proceeding with the program; a microprogram for restoring a set value without using the evacuated information necessary in proceeding with the program; and a program for resetting processes, and

Inter Partes Review of U.S. Patent No. 7,930,575 B2

		the CPU selects one of the plurality of microprograms in an alternative way.
Claim 26	--	The microcontroller according to claim 17, further comprising a second power supply unit for lowering a voltage to be supplied to a circuit portion outside a power shutdown range to a lower limit voltage for information holding when the power shutdown factor occurs.
Claim 30	--	The microcontroller of claim 17, wherein the power supply unit includes a voltage regulator.
Claim 31	--	The microcontroller of claim 17, wherein the power supply unit does not shut down the power supply at least to the clock generator upon receiving the power shutdown control signal from the power supply control unit.