

UNITED STATES PATENT AND TRADEMARK OFFICE

---

**BEFORE THE PATENT TRIAL AND APPEAL BOARD**

---

SAMSUNG ELECTRONICS CO., LTD.,  
SAMSUNG ELECTRONICS AMERICA, INC., AND  
QUALCOMM INCORPORATED,  
Petitioner

v.

DAEDALUS PRIME LLC,  
Patent Owner.

---

Case No. IPR2023-00550  
U.S. Patent No. 8,775,833

---

**PETITION FOR *INTER PARTES* REVIEW  
OF U.S. PATENT NO. 8,775,833**

## TABLE OF CONTENTS

	Page
I. INTRODUCTION .....	1
II. MANDATORY NOTICES UNDER 37 C.F.R. §42.8.....	2
III. FEE AUTHORIZATION .....	5
IV. GROUNDS FOR STANDING.....	5
V. PRECISE RELIEF REQUESTED .....	5
VI. THE CHALLENGED PATENT .....	6
VII. PATENT PROSECUTION HISTORY .....	8
VIII. LEVEL OF ORDINARY SKILL IN THE ART.....	8
IX. PRIORITY DATE .....	8
X. CLAIM CONSTRUCTION .....	9
XI. BRIEF DESCRIPTION OF THE APPLIED PRIOR ART REFERENCES .....	9
A. Finkelstein (Ex-1005).....	9
B. Conroy (Ex-1006).....	11
C. Nussbaum (Ex-1007).....	13
D. Bose (Ex-1008).....	15
XII. DETAILED EXPLANATION OF THE UNPATENTABILITY GROUNDS .....	16
A. Ground 1: Claims 1-5 and 7 are rendered obvious by Finkelstein. ....	16
1. Independent claim 1 .....	16
2. Dependent claims 2-5 and 7.....	26
B. Ground 2: Claim 7 is rendered obvious by Finkelstein in view of Conroy.....	33
1. Dependent claim 7 .....	33
C. Ground 3: Claims 13-15 and 17-18 are rendered obvious by Nussbaum. ....	35

**TABLE OF CONTENTS**  
(continued)

	<b>Page</b>
1. Independent claim 13 .....	35
2. Dependent claims 14-15 and 17-18 .....	45
D. Ground 4: Claim 16 is rendered obvious by Nussbaum in view of Bose.....	50
1. Dependent claim 16 .....	50
E. Ground 5: Claims 13-15 and 17-18 are rendered obvious by Conroy. ....	54
1. Independent claim 13 .....	54
2. Dependent claims 14-15 and 17-18 .....	63
F. Ground 6: Claim 16 is rendered obvious by Conroy in view of Bose. ....	66
1. Dependent claim 16 .....	66
G. Ground 7: Claims 1-5, 7, 13-15, and 17-18 are rendered obvious by Conroy in view of Finkelstein. ....	68
1. A POSITA would be motivated to combine the teachings of Conroy and Finkelstein, and would have a reasonable expectation of success in doing so. ....	68
2. Independent claim 1 .....	70
3. Dependent claims 2-5 and 7.....	80
4. Claims 13-15 and 17-18.....	83
XIII. THERE IS NO BASIS FOR ANY DISCRETIONARY DENIAL.....	84
XIV. CONCLUSION.....	84

**LIST OF EXHIBITS<sup>1</sup>**

Ex-1001	U.S. Patent No. 8,775,833
Ex-1002	Declaration of Dr. Trevor Mudge
Ex-1003	Curriculum Vitae of Dr. Trevor Mudge
Ex-1004	Prosecution History of U.S. Patent No. 8,775,833
Ex-1005	U.S. Published Patent Application No. 2010/0115304 (“Finkelstein”)
Ex-1006	U.S. Published Patent Application No. 2007/0049133 (“Conroy”)
Ex-1007	U.S. Published Patent Application No. 2011/0022356 (“Nussbaum”)
Ex-1008	U.S. Published Patent Application No. 2009/0089602 (“Bose”)
Ex-1009	INTENTIONALLY LEFT BLANK
Ex-1010	INTENTIONALLY LEFT BLANK
Ex-1011	INTENTIONALLY LEFT BLANK
Ex-1012	INTENTIONALLY LEFT BLANK
Ex-1013	INTENTIONALLY LEFT BLANK
Ex-1014	INTENTIONALLY LEFT BLANK
Ex-1015	INTENTIONALLY LEFT BLANK
Ex-1016	INTENTIONALLY LEFT BLANK
Ex-1017	INTENTIONALLY LEFT BLANK
Ex-1018	INTENTIONALLY LEFT BLANK
Ex-1019	INTENTIONALLY LEFT BLANK

---

<sup>1</sup> Four-digit pin citations that begin with 0 are to the page stamps added by Samsung in the bottom right corner of the exhibits. All other pin citations are to original page, column, paragraph, and/or line numbers.

Ex-1020	Claim Mapping Table
Ex-1021	INTENTIONALLY LEFT BLANK
Ex-1022	INTENTIONALLY LEFT BLANK
Ex-1023	U.S. Patent No. 7,263,457 (“White”)
Ex-1024	U.S. Patent No. 8,438,416 (“Kocev”)
Ex-1025	U.S. Patent No. 8,984,523 (“Vajda”)
Ex-1026	U.S. Patent No. 7,779,276 (“Bolan”)
Ex-1027	Selected pages of Stefanos Kaxiras et al., <i>Computer Architecture Techniques for Power-Efficiency</i> (2008)
Ex-1028	U.S. Published Patent Application No. 2010/0162006 (“Therien”)

## I. INTRODUCTION

Samsung Electronics Co., Ltd., Samsung Electronics America, Inc., and Qualcomm Incorporated (“Petitioner”) requests *inter partes* review (“IPR”) of Claims 1-5, 7, and 13-18 of U.S. Patent No. 8,775,833 (“the ’833 Patent”) (Ex-1001), currently assigned to Daedalus Prime LLC (“Patent Owner”).

The ’833 Patent relates to “determining a power budget for a multi-domain processor for a current time interval.” Ex-1001, Abstract. In particular, the ’833 Patent states that an issue in the field is that “processor designs have evolved ... to multicore processors that include multiple processor cores within a single IC package.” *Id.* at 1:11-26. Over time, “ever greater numbers of cores and related circuitry are being incorporated into processors and other semiconductors.” *Id.* But the ’833 Patent alleges “suitable mechanisms to ensure that these different units have sufficient power do not presently exist.” *Id.* The ’833 Patent attempts to address this purported issue.

However, the prior art in this Petition demonstrates that Claims 1-5, 7, and 13-18 of the ’833 Patent involve well-known design choices in the art used to address the well understood issue of allocating power to multiple processor cores. Thus, Claims 1-5, 7, and 13-18 of the ’833 Patent would have been obvious over the prior art.

This Petition presents non-cumulative grounds of unpatentability not addressed during patent prosecution, as discussed further in Section XII below. The grounds presented in this Petition are reasonably likely to prevail, this Petition should be granted, a trial should be instituted, and the challenged claims should be cancelled.

## II. MANDATORY NOTICES UNDER 37 C.F.R. §42.8

**Real Parties-in-Interest:** Petitioner identifies the following real parties-in-interest: Samsung Electronics Co., Ltd., Samsung Electronics America, Inc., Qualcomm Incorporated, and Qualcomm Technologies, Inc.

Petitioner Qualcomm Incorporated further identifies, without conceding any such party is a Real Party-in-Interest, Mazda Motor Corporation, Mazda North American Operations, Mazda Motor of America, Inc., Mercedes-Benz Group AG, Mercedes-Benz AG, Mercedes-Benz USA, LLC, and Visteon Corporation as parties that Patent Owner has accused of infringing the '833 Patent based on incorporation of Qualcomm products in *Daedalus Prime LLC v. Mazda Motor Corporation et al.*, 1:22-cv-01109 (D. Del.) and *Certain Semiconductors and Devices and Products Containing the Same, Including Printed Circuit Boards, Automotive Parts, and Automobiles*, ITC Inv. No. 337-TA-1332 (Violation).

**Related Matters:** Patent Owner has asserted the '833 Patent against Petitioner in (1) *Daedalus Prime LLC v. Arrow Electronics, Inc., et al.*, 1:22-cv-

01107 (D. Del.); (2) *Daedalus Prime LLC v. Mazda Motor Corporation, et al.*, 1:22-cv-01109 (D. Del.); (3) *Daedalus Prime LLC v. Mazda Motor Corporation, et al.*, 1:22-cv-01108 (D. Del.); (4) *Daedalus Prime LLC v. Samsung Electronics Co., Ltd., et al.*, 2:22-cv-00352 (E.D. Tex.); (5) *Certain Integrated Circuits, Mobile Devices Containing the Same, and Components Thereof*, U.S. International Trade Commission (“ITC”) Inv. No. 337-TA-1335 (Violation); (6) *Certain Semiconductors and Devices and Products Containing the Same, Including Printed Circuit Boards, Automotive Parts, and Automobiles*, ITC Inv. No. 337-TA-1332 (Violation).

**Lead and Back-Up Counsel:**

- Lead Counsel:

William M. Fink (Reg. No. 72,332)  
O’Melveny & Myers LLP  
1625 Eye Street, NW  
Washington, DC 20006  
Telephone: (202) 383-5300  
Fax: (202) 383-5414  
E-Mail: [tfink@omm.com](mailto:tfink@omm.com)

- Backup Counsel:

Benjamin Haber (Reg. No. 67,129)  
Nicholas J. Whilt (Reg. No. 72,081)  
Brian Cook (Reg. No. 59,356)  
O’Melveny & Myers LLP  
400 South Hope Street, 18th Floor  
Los Angeles, CA 90071  
Telephone: (213) 430-6000  
Fax: (213) 430-6407



E-Mail: [bhaber@omm.com](mailto:bhaber@omm.com)

E-Mail: [nwhilt@omm.com](mailto:nwhilt@omm.com)

E-Mail: [bcook@omm.com](mailto:bcook@omm.com)

*Counsel for Petitioner Samsung Electronics Co., Ltd. and  
Samsung Electronics America, Inc.*

Daniel Leventhal (Reg. No. 59,576)

Richard Zembek (Reg. No. 43,306)

Darren Smith (Reg. No. 64,261)

Norton Rose Fulbright

1301 McKinney, Suite 5100

Houston, TX 77010

Telephone: (713) 651-5151

Fax: (713) 651-5246

E-Mail: [daniel.leventhal@nortonrosefulbright.com](mailto:daniel.leventhal@nortonrosefulbright.com)

E-Mail: [richard.zembek@nortonrosefulbright.com](mailto:richard.zembek@nortonrosefulbright.com)

E-Mail: [darren.smith@nortonrosefulbright.com](mailto:darren.smith@nortonrosefulbright.com)

*Counsel for Petitioner Qualcomm Incorporated*

Eagle H. Robinson (Reg. No. 61,361)

Norton Rose Fulbright

98 San Jacinto Blvd., Ste. 1100

Austin, TX 78701

Telephone: (512) 474-5201

Fax: (512) 536-4598

E-Mail: [eagle.robinson@nortonrosefulbright.com](mailto:eagle.robinson@nortonrosefulbright.com)

*Counsel for Petitioner Qualcomm Incorporated*

**Service Information:** Petitioner consents to electronic service by email to the  
following addresses:

- [tfink@omm.com](mailto:tfink@omm.com)
- [bhaber@omm.com](mailto:bhaber@omm.com)
- [nwhilt@omm.com](mailto:nwhilt@omm.com)
- [bcook@omm.com](mailto:bcook@omm.com)
- [daniel.leventhal@nortonrosefulbright.com](mailto:daniel.leventhal@nortonrosefulbright.com)
- [richard.zembek@nortonrosefulbright.com](mailto:richard.zembek@nortonrosefulbright.com)
- [darren.smith@nortonrosefulbright.com](mailto:darren.smith@nortonrosefulbright.com)

- [eagle.robinson@nortonrosefulbright.com](mailto:eagle.robinson@nortonrosefulbright.com)

### III. FEE AUTHORIZATION

Pursuant to 37 C.F.R. §42.15(a) and §42.103(a), the PTO is authorized to charge any and all fees to Deposit Account No. LA500639.

### IV. GROUNDS FOR STANDING

Petitioner certifies that the '833 Patent is available for review, this Petition is timely filed, and Petitioner is not barred or estopped from requesting review.

### V. PRECISE RELIEF REQUESTED

Petitioner requests review and cancellation of Claims 1-5, 7, and 13-18 as unpatentable based on the following grounds, supported by a declaration from Dr. Trevor Mudge. Ex-1002 ¶¶64-255.

Ground	Summary
1	Claims 1-5 and 7 are obvious over Finkelstein (Ex-1005).
2	Claim 7 is obvious over Finkelstein (Ex-1005) in view of Conroy (Ex-1006).
3	Claims 13-15 and 17-18 are obvious over Nussbaum (Ex-1007).
4	Claim 16 is obvious over Nussbaum (Ex-1007) in view of Bose (Ex-1008).
5	Claims 13-15 and 17-18 are obvious over Conroy (Ex-1006).

Ground	Summary
6	Claim 16 is obvious over Conroy (Ex-1006) in view of Bose (Ex-1008).
7	Claims 1-5, 7, 13-15, and 17-18 are obvious over Conroy (Ex-1006) in view of Finkelstein (Ex-1005).

## VI. THE CHALLENGED PATENT

The '833 Patent is directed to “a power budget of a processor including multiple domains [that] can be dynamically apportioned at run time.” Ex-1001, 1:56-58. In particular, the '833 Patent is directed to allocating power in multicore processors based on the respective core workload. *Id.* at 1:20-26. The patent explains that such processors can be divided into multiple “domains” (also referred to as “planes”) where each domain is a “collection of hardware and/or logic that operates at the same voltage and frequency point.” *Id.* at 1:56-60, 4:7-21. As shown in Figure 7, a multidomain processor may include “a plurality of cores 410a-410n [red], a graphics domain 420 [purple], and a system agent domain 450 [yellow].” *Id.* at 9:13-16.

400

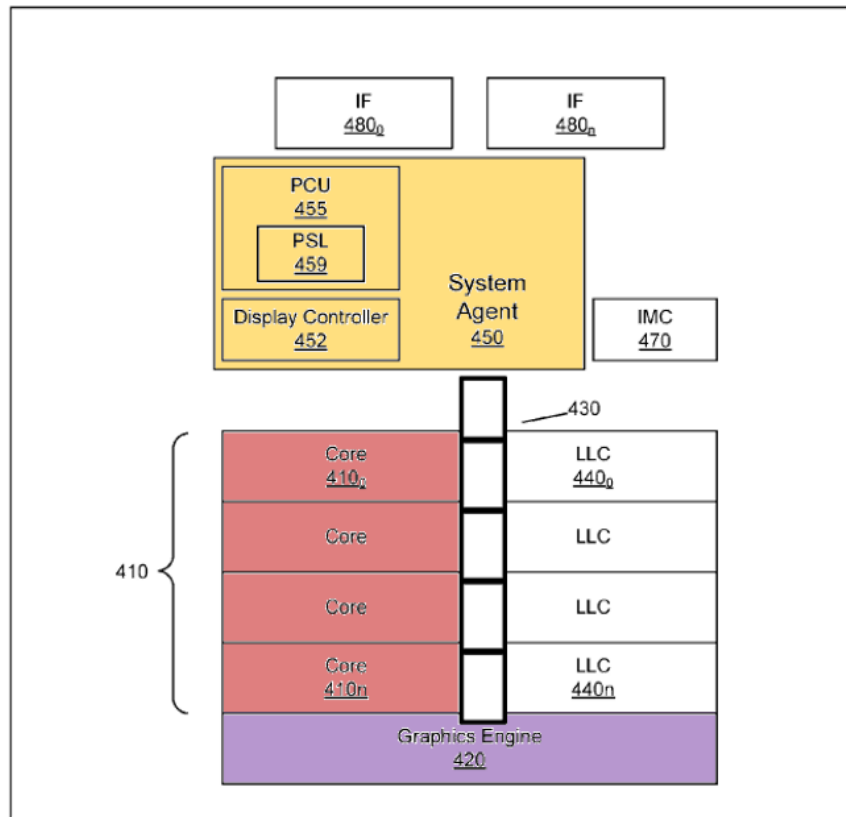


FIG. 7

The '833 Patent describes how multiple domains may share a dynamically re-partitioned power budget, and “[a]s a result, a power budget or power headroom can be reallocated between cores and graphics engine when they are both integrated on the same die.” Ex-1001, 3:21-26, 8:26-31, 10:4-24. The '833 Patent further explains: “a user may be provided with control, e.g., by user-level software to enable the software to determine how package power budget is shared between different domains.” *Id.* at 4:1-4.

As explained in detail below, the '833 Patent claims are obvious in view of the prior art.

## **VII. PATENT PROSECUTION HISTORY**

The '833 Patent file history is submitted as Ex-1004. The application leading to the '833 Patent was filed on February 28, 2013. It purports to be a continuation of U.S. patent application serial no. 13/225,677, filed on September 6, 2011.

On February 5, 2014, Applicant responded to non-final §§ 101 and 102 rejections, and amended the claims to their present scope. On February 18, 2014, the Examiner allowed the claims, and the '833 Patent issued on July 8, 2014.

## **VIII. LEVEL OF ORDINARY SKILL IN THE ART**

A person of ordinary skill in the art at the relevant time (“POSITA”) would have had a bachelor’s degree in electrical engineering, computer science, computer engineering, material science, physics, applied physics, or a related field, and at least two years of experience in the research, design, development, or testing of electronic circuits or components or software for controlling electronic circuits or components, or the equivalent, with additional education substituting for experience and vice versa. Ex-1002 ¶¶25, 37-43.

## **IX. PRIORITY DATE**

The earliest possible priority for the '833 Patent is September 6, 2011. Petitioner takes no position on the proper priority date for each claim of the '833

Patent. Assuming this earliest date, all prior art references asserted in this petition are prior art.

## **X. CLAIM CONSTRUCTION**

Petitioner interprets the claims of the '833 Patent according to 37 C.F.R. § 42.100(b). To resolve the particular grounds presented in this Petition, Petitioner does not believe that any term requires explicit construction. Ex-1002 ¶62.<sup>2</sup>

## **XI. BRIEF DESCRIPTION OF THE APPLIED PRIOR ART REFERENCES**

### **A. Finkelstein (Ex-1005)**

Finkelstein, titled “Power management for multiple processor cores,” is U.S. Published Patent Application No. 2010/0115304, published on May 6, 2010. Finkelstein is prior art under at least pre-AIA 35 U.S.C. § 102(b). Like the '833 Patent, Finkelstein's issued patent was also developed by and assigned to Intel. Additionally, Finkelstein shares two common inventors with the '833 Patent: Efraim Rotem and Doron Rajwan.

Just like the '833 Patent, Finkelstein “relates to power management for multiple processor cores.” Ex-1005 ¶1. As shown in annotated Figure 1, Finkelstein

---

<sup>2</sup> Claim construction procedures have not yet begun in the related district court or ITC matters. Petitioner reserves the right to revisit and revise these constructions in district court and the ITC as necessary. Additionally, Petitioner will request leave to submit any claim construction order as soon as it becomes available, so that it is timely made of record in the proceeding and can be considered by the Board. 37 C.F.R. § 42.100(b).

discloses a system that includes “one or more processors” (purple rectangle) where each processor “may include one or more processor cores” (red). *Id.* ¶¶10-11.

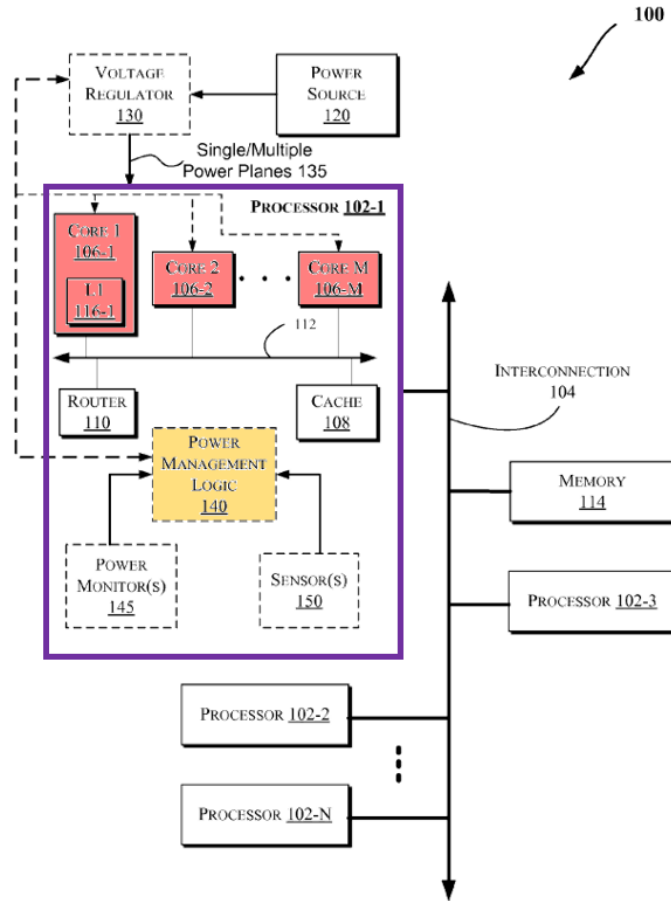


FIG. 1

Each processor may communicate with various other components, including memory, voltage regulator(s), and power source(s). Ex-1005 ¶¶13-14. A voltage regulator “may be coupled to a single power plane ... or to multiple power planes ... where each power plane may supply power to a different core or group of cores.” *Id.* ¶14.

Finkelstein also discloses “a power management logic 140 [yellow, above] to control supply of power to” processor components. Ex-1005 ¶¶16, 24, 28-36. Logic 140 “may request the cores 106 to modify their operating frequency, power consumption, etc.” *Id.* ¶16. It may also “consider statistics ... on operating states of the cores” and “select[] technique(s) [that] may be applied to throttle or modify an operational characteristic of one or more of the cores.” *Id.* ¶24. Finkelstein also discloses that “a controller budget may be defined (e.g., implemented by the logic 140)” and provides several formulae by which logic 140 may dynamically allocate the power budget between power planes. *Id.* ¶¶28-36. For example, Finkelstein discloses sharing an “energy budget” so that “unused processor core power” can be used for other cores that may be performing more “intensive workload.” *See, e.g., id.* ¶26. Finkelstein also discloses that “user (or application defined) preferences” may be considered in determining power budget allocation. *Id.* ¶34; *see also id.* ¶¶29, 31.

**B. Conroy (Ex-1006)**

Conroy, titled “Methods and apparatuses for dynamic power control,” is U.S. Published Patent Application No. 2007/0049133, published on March 1, 2007. Conroy qualifies as prior art under at least pre-AIA 35 U.S.C. § 102(b).

Conroy discloses “dynamically redistribut[ing] power in a [computer] system that includes a plurality of subsystems.” Ex-1006, Abstract. For example, Conroy



teaches identifying a “load profile of the system” and redistributing power “between the subsystems while tracking the load profile.” *Id.* Conroy discloses that its “subsystems may be, for example, a central processing unit (‘CPU’) and a graphics processor (‘GPU’),” and that a “power weighting arrangement between the subsystems may be a power distribution table selected out of a plurality of power distribution tables” that describe, among other things, “power used by each of the subsystems at various system’s settings.” *Id.* ¶¶5-6. Examples of power distribution tables are illustrated in Figures 21A-C and 22, copied below.

Balanced load profile K1 (e.g., 0.5, 0.5)

2100	2101	2102	2103	2104	Fig. 21 A
2105	Settings	CPU	GPU	Allocated Power	
	Setting 1	A1 (e.g., 10)	B1 (e.g., 10)	P1 (e.g., 20)	
	Setting 2	A2 (e.g., 7)	B2 (e.g., 7)	P2 (e.g., 14)	
	Setting 3	A3 (e.g., 5)	B3 (e.g., 5)	P3 (e.g., 10)	

CPU-heavy load profile K2 (e.g., 0.75, 0.25)

2110	2101	2102	2103	2104	Fig. 21 B
2105	Settings	CPU	GPU	Allocated Power	
	Setting 1	A1 (e.g., 15)	B1 (e.g., 5)	P1 (e.g., 20)	
	Setting 2	A2 (e.g., 10)	B2 (e.g., 4)	P2 (e.g., 14)	
	Setting 3	A3 (e.g., 7)	B3 (e.g., 3)	P3 (e.g., 10)	

GPU-heavy load profile K3 (e.g., 0.25, 0.75)

2120	2101	2102	2103	2104	Fig. 21 C
2105	Settings	CPU	GPU	Allocated Power	
	Setting 1	A1 (e.g., 5)	B1 (e.g., 15)	P1 (e.g., 20)	
	Setting 2	A2 (e.g., 3.5)	B2 (e.g., 10)	P2 (e.g., 14)	
	Setting 3	A3 (e.g., 3)	B3 (e.g., 7)	P3 (e.g., 10)	

Load Profile Kn

2208 {	2201	2202	2203	2204	2205	2206
	Settings	Subsystem 1	Subsystem 2	Subsystem...	Subsystem N	Power
	1	A1	B1	...	C1	P1
	2	A2	B2	...	C2	P2
	M	Am	Bm	...	Cm	Pm

2209

**Fig. 22**

Conroy’s power distribution table may correspond to various power distribution profiles, including where one subsystem requires most of the power followed by another system requiring most of the power. For example, during a “CPU-heavy load profile,” when CPU workload is intensive, power may be distributed such that “CPU consumes about 75%, while GPU consumes about 25% of the power allocated for the system.” Ex-1006 ¶203. And, if another subsystem, such as the GPU, is engaged in intensive workload after the CPU, “the power of the system is shifted towards GPU” such that, for example, “GPU consumes about 75%, while CPU consumes about 25% of the power allocated for the system.” *Id.*

### **C. Nussbaum (Ex-1007)**

Nussbaum, titled “Determining performance sensitivities of computational units,” is U.S. Published Patent Application No. 2011/0022356, filed on July 24, 2009, and published on January 27, 2011. Nussbaum qualifies as prior art under at least pre-AIA 35 U.S.C. §§ 102(a) and 102(e).

Nussbaum discloses power management of computational units in a “System on a Chip (SOC)” and allocation of “power headroom” to various “on-die or on-platform components.” Ex-1007 ¶20, Abstract. Nussbaum also discloses “reallocation of the power from idle or less active components to the busy components by having more power allocated to the busy ones.” *Id.* ¶23. Nussbaum explains that, for example, “one way to determine how to allocate power between Core0 and Core1 ... is to know which of the two cores, if any, can better exploit an increase in performance capability provided, e.g., by an increase in frequency.” *Id.* ¶25. Nussbaum also explains that “performance sensitivity of each computational unit to frequency change, and/or other change in performance capability, also referred to herein as boost sensitivity, is determined and stored on a computational unit basis.” *Id.*

Nussbaum further discloses dynamically allocating the power budget according to whether an application is “CPU-bounded or GPU-bounded.” Ex-1007 ¶47. If an application is CPU-bounded, cores that are “less sensitive in terms of performance to a reduction in performance capability” are selectively throttled so that the “power headroom” can be reallocated to the computational unit(s) executing the CPU-bounded operation. *Id.* If an application is GPU-bounded, the “power headroom” may be reallocated to the unit(s) executing the GPU-bounded operation. *Id.*

**D. Bose (Ex-1008)**

Bose, titled “Method and system of peak power enforcement via autonomous token-based control and management,” is U.S. Published Patent Application No. 2009/0089602, published on April 2, 2009. Bose qualifies as prior art under at least pre-AIA 35 U.S.C. § 102(b).

Bose discloses “power management of a system of connected components ... wherein each component is assigned a power budget as determined by a number of allocated tokens in the token allocation map.” Ex-1008, Abstract. Bose utilizes a “token management unit (TMU)” to manage allocation of tokens to “token-managed units,” such as “processor resources,” and explains that “[e]ach managed component, in response to its own operating conditions (as evidenced by its utilization markers) can either attempt to donate one or more token(s)” or “read (receive) one or more token(s).” *Id.* ¶¶38-39, 68. “Each token connotes a pre-specified quantum of power, and as such, the total number of tokens owned by a component determines its own power limit.” *Id.* ¶68.

Bose also describes an implementation in which the “TMU fixes a budget of 3 tokens for all other non-managed units (including itself), and distributes the remaining 12 tokens across the 4 managed units.” Ex-1008 ¶53. Bose further discloses that if the performance level of a particular resource is “below a predefined minimum performance,” a process is initiated to “try and read additional tokens from

[a] centralized bus token storage ... to boost up performance” of that particular resource, e.g., “by increasing its voltage and frequency, if allowed by the system architecture.” *Id.* ¶86, Fig. 13. Bose explains that if no additional tokens can be allocated to the particular resource at the moment, that particular resource can wait for its token count to be increased in the next iteration of the process. *Id.*

## **XII. DETAILED EXPLANATION OF THE UNPATENTABILITY GROUNDS**

### **A. Ground 1: Claims 1-5 and 7 are rendered obvious by Finkelstein.**

#### **1. Independent claim 1**

##### **a. Element 1[Pre]: A processor comprising:**

Finkelstein teaches the preamble of Claim 1, to the extent it is limiting.  
Ex-1002 ¶¶64-66.

An example of Finkelstein’s processor 102-1 (purple) is shown in annotated Figure 1 below.

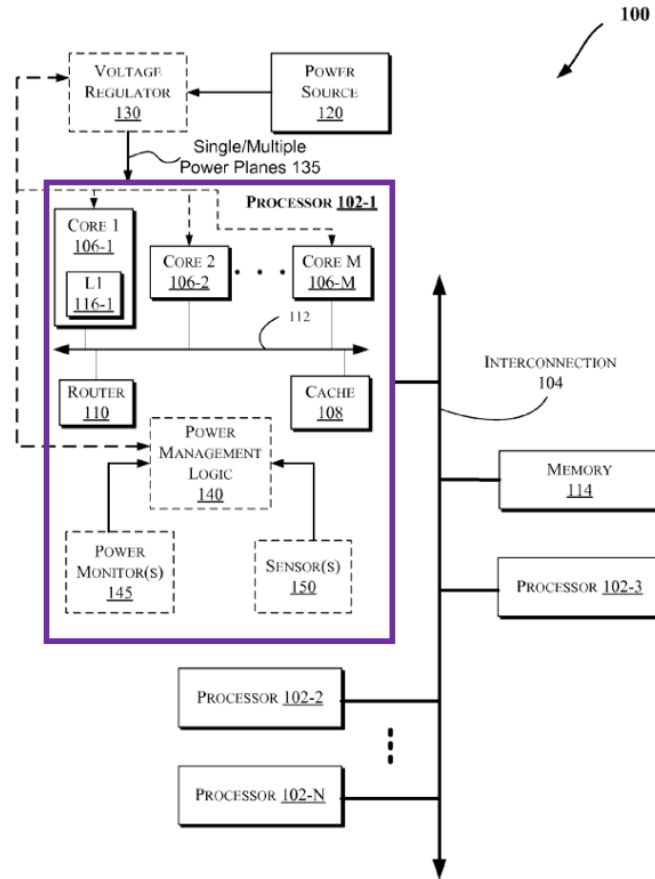


FIG. 1

- b. **Element 1[A]: a first domain and a second domain, each of the first and second domains to operate at an independent voltage and frequency;**

Finkelstein teaches Element 1[A]. Ex-1002 ¶¶67-71.

Finkelstein teaches a first domain and a second domain. As shown in annotated Figure 1 below, Finkelstein discloses cores in “multiple power planes 135 (e.g., where each power plane may supply power to a different core or group of cores).” Ex-1005 ¶14. A POSITA would have understood that each “power plane” 135 corresponds to a “domain.” Ex-1002 ¶68. Indeed, the ’833 Patent itself refers to “domains” as “planes.” Ex-1001, 4:7-21. Thus, a POSITA would have

understood that a first set of processor cores, e.g., Core 1 (red) in a first power plane, is a “first domain” and another set of cores, e.g., Core 2 (blue) in a second power plane, is a “second domain.” Ex-1002 ¶168.

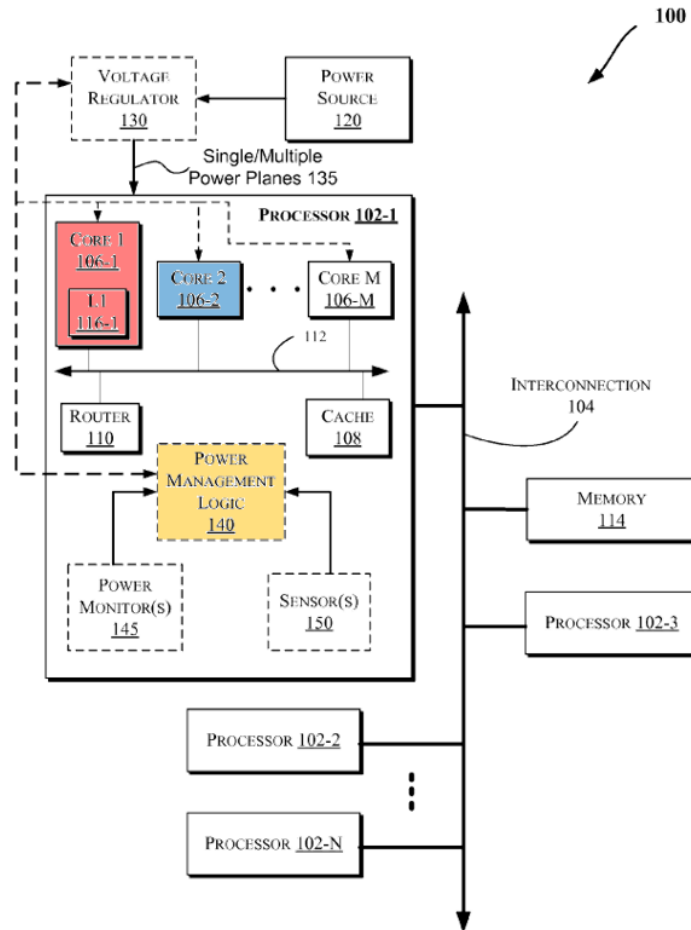


FIG. 1

Finkelstein further discloses that each of the first and second domains operates at an independent voltage and frequency. Finkelstein uses “power management logic 140 to control supply of power to components of the processor 102 (e.g., cores 106),” and power management logic 140 (above, yellow) “may request the cores [] to modify their operating frequency, power consumption, etc.” Ex-1005 ¶16; *see also*

*id.* ¶31. Logic 140 can also “throttle or modify an operational characteristic of ***one*** or more of the cores 106 (such as an operating voltage and/or an operating frequency of ***a processor core*** 106).” *Id.* ¶24.<sup>3</sup> A POSITA implementing Finkelstein would thus have been motivated to operate Core 1 (in a first domain) and Core 2 (in a second domain) at independent voltages and frequencies by controlling the “operational characteristic of one” of the cores such that logic 140 could modify the operating voltage and/or frequency of either Core 1 or Core 2, without affecting the operating voltage and/or frequency of the other. *Id.*; *see also id.* ¶¶27, 32, 39 (discussing control of power budget “per power plane”); Ex-1002 ¶69. This independent operation would have allowed for a greater degree of control over one core, while still allowing for increased performance on the other. Ex-1002 ¶69. For example, one core that was operating too hot could be set to a lower frequency, while the other core could continue to operate at a higher frequency and performance. *Id.*; *see also id.* ¶70 (further describing Figure 1, which depicts mechanisms that facilitate independent control of voltage and frequency to different domains).

**c. Element 1[B]: a memory controller coupled to the first and second domains;**

Finkelstein teaches Element 1[B]. Ex-1002 ¶¶72-75.

---

<sup>3</sup> Throughout this Petition, all annotations and emphasis have been added, unless otherwise noted.



Finkelstein discloses that processor cores “may be implemented on a single integrated circuit (IC) chip,” which in turn may include “memory controllers (such as those discussed with reference to FIGS. 6-7).” Ex-1005 ¶11. “Moreover, various components of the processor 102-1 may communicate with the cache 108 directly, through a bus (e.g., the bus 112), **and/or a memory controller or hub.**” *Id.* ¶13. An example of this is shown in annotated Figure 7 (referred to in the description of Figure 1), which depicts processors 702 and 704, each including “a local memory controller hub (MCH)” (green). *Id.* ¶49. Finkelstein also explains that processors 702 and 704, each including “a local memory controller hub (MCH),” may include “one or more of the cores 106, logic 140, sensor(s) 150, and/or power monitor(s) 145 of FIG. 1.” *Id.* Therefore, Finkelstein teaches a memory controller coupled to a core in the first domain (e.g., Core 1) and a core in the second domain (e.g., Core 2).

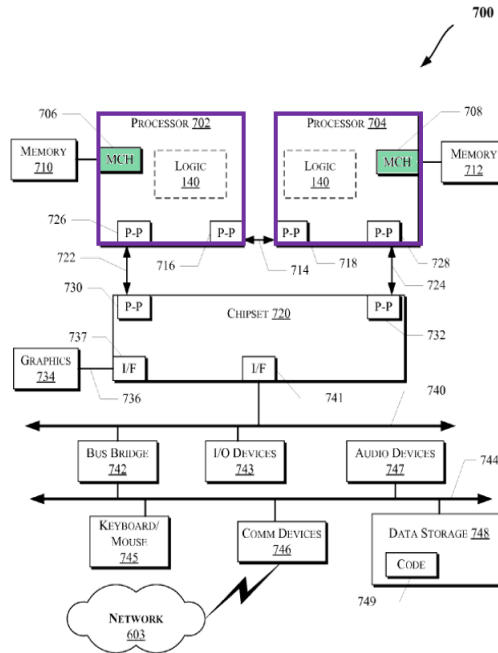


Fig. 7

**d. Element 1[C]: at least one interface; and**

Finkelstein teaches Element 1[C]. Ex-1002 ¶¶76-78.

As illustrated in annotated Figure 1, Finkelstein discloses that processor cores “may be implemented on a single integrated circuit (IC) chip,” which in turn may include “buses or interconnections (such as a bus or interconnection 112),” and “various components of the processor 102-1” may communicate with cache 108 and router 110 through interconnection 112 (green). Ex-1005 ¶¶11-13, Fig. 1. A POSITA would have understood Finkelstein as teaching an “interface” because an “interconnection” for communications between various components of processor 102-1, cache 108, and router 110 serves as an interface between these components. *Id.* ¶¶10-13; Ex-1002 ¶77. Indeed, Finkelstein itself discloses using “interfaces” to

“interconnect[]” processors, memory, and input/output devices, and a POSITA would have understood to implement interconnection 112 (green) using the same or similar “interfaces.” *Id.* ¶¶47-48; Ex-1002 ¶77.

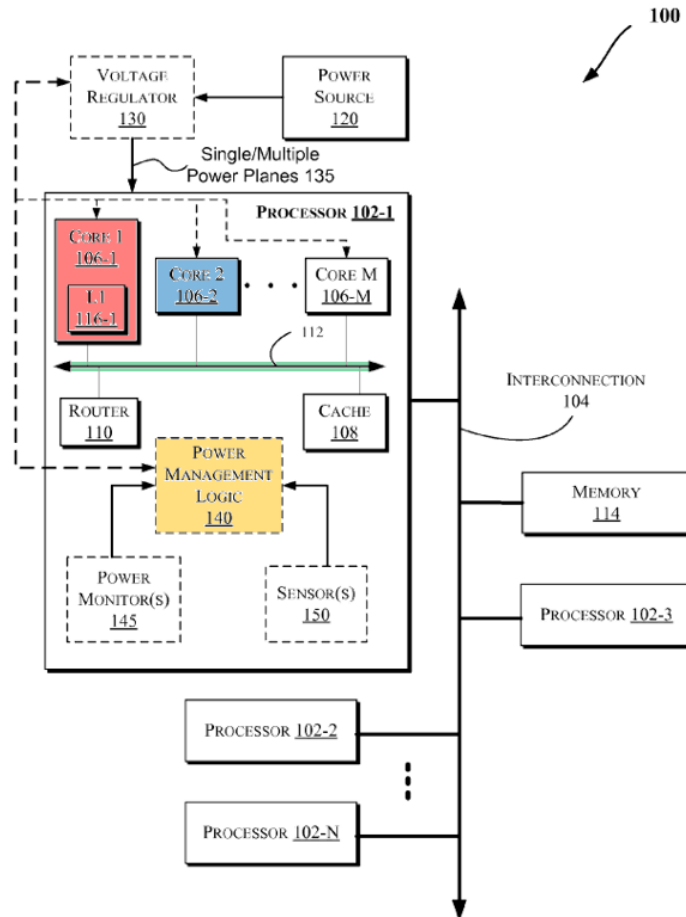


FIG. 1

- e. **Element 1[D.i]: first logic to dynamically allocate a power budget for the processor between the first and second domains at run time**

Finkelstein teaches Element 1[D.i]. Ex-1002 ¶¶79-81.

Finkelstein discloses that “if multiple power planes [are] sharing the same power source,” then these planes may “share a common package power/energy

budget” so that, for example, “unused processor core power” of a core in the first domain (e.g., Core 1) can be used for “more Graphics Effect(s) (GFX) performance in a GFX intensive workload” being performed by a core in the second domain (e.g., Core 2). Ex-1005 ¶26. Thus, a POSITA would have recognized that Finkelstein teaches allocating a power budget between the first and second domains “at run time,” while the GFX intensive workload is being performed. *Id.*; Ex-1002 ¶80. Moreover, Finkelstein discloses implementations in which an “unused” budget for domain  $i$ , defined as a variable  $W_{ki}E^k - E_{high}^i$ , is “distributed among the rest of the power planes [domains] proportionally to their weights,” defined as variables  $W_{ki}$ . Ex-1005 ¶35. A POSITA would have recognized such an allocation, where the power budget itself is a variable ( $W_{ki}E^k - E_{high}^i$ ) and the variable power budget is being distributed based on variable weights ( $W_{ki}$ ), as a “dynamic” allocation because the allocation can change during run-time to correspond to the then-present operating conditions. *Id.*; *see also id.* ¶¶19, 22, 24, 26; Ex-1002 ¶80. Furthermore, Finkelstein explains that its power allocation framework handles changes “on-the-fly” and takes into account a “decay component” that “corresponds to [a] window size of seconds [or] much smaller,” confirming that Finkelstein allocates power “dynamically” to correspond to the then-present operating conditions. *Id.* ¶¶29-31; Ex-1002 ¶80; *see also* Section XII.A.1.f.

- f. **Element 1[D.ii]: according to a first sharing policy value for the first domain and a second sharing policy value for the second domain, the first and second sharing policy values controllable by user-level software.**

Finkelstein teaches Element 1[D.ii]. Ex-1002 ¶¶82-86.

Finkelstein discloses two complimentary approaches to dynamically allocate a power budget according to a first sharing policy value for the first domain and a second sharing policy value for the second domain, wherein the sharing policies are controllable by user-level software.

(1) First Approach

Finkelstein discloses a set of energy sharing policy values for each plane “i,” expressed as vectors  $W_{ki}$ , in the following equation used to determine  $E^{k,i}$ , which represents the portion of the power budget plane “i” may obtain:

$$E^{k,i} = W_{ki} E^k \quad (2)$$

Finkelstein explains that “[f]or constraint k, *user (or application defined) preferences* that describe how budget  $E^k$  is *distributed among power planes* may be determined.” Ex-1005 ¶34. Finkelstein also explains that “such information may be provided as an input in the form of vectors  $W_k$  of length m, *such that entry i corresponds to the portion of the budget that goes to power plane i.*” *Id.* Thus, in equation (2),  $W_{k1}$  represents a first sharing policy value for the first domain (receiving power from power plane  $i=1$ ) and  $W_{k2}$  represents a second sharing policy

value for the second domain (receiving power from power plane i=2). Furthermore, Finkelstein explains that “unused” budget for a particular power plane “may be distributed among the rest of the power planes proportionally to” these sharing policy values,  $W_{k1}$  and  $W_{k2}$  (referred to as weights), and that these sharing policy values are controllable by user-level software (because they are user preferences and a POSITA would have understood that such preferences are provided by users via user-level software, e.g., via a graphical interface). *Id.* ¶¶34-35; Ex-1002 ¶84.

(2) Second Approach

Finkelstein also discloses a set of sharing policy values based on Thermal Design Power (TDP) power limit. Finkelstein explains that “energy budget may be managed and/or power setting(s) (e.g., voltage and/or frequency changes) may be made accordingly [*sic*] to a current budget,” and that “[e]nergy-based power management may be performed by controlling the energy budget defined iteratively as:”

$$E_{n+1} = \alpha E_n + (TDP_n - P_n) \Delta t_n \quad (1)$$

Ex-1005 ¶28. In equation (1), “ $TDP_n$  is the Thermal Design Power (TDP) power limit on step n,” “ $P_n$  is the power spent on step n over time  $\Delta t_n$ ,” “ $\alpha$  is the decay component,” and “ $E_n$  corresponds to the energy ‘remainder’ which is the amount of energy not consumed by the system.” *Id.* ¶29. Finkelstein explains that “the system (e.g., logic 140) may set ***TDP constraints on each of the power planes separately,***”

and notes that “this framework smoothly handles a case when ***TDP is changed on-the-fly (e.g., by a software application or a user).***” *Id.* ¶31. Thus, in equation (1), TDP constraints on a first power plane (the power plane that supplies power to the first domain) represents a first sharing policy value and TDP constraints on a second power plane (the power plane that supplies power to the second domain) represents a second sharing policy value. Furthermore, these sharing policy values, i.e., TDP constraints on each of the power planes, are controllable by user-level software because they can be “changed on-the-fly” by a user and a POSITA would have understood that such changes are made via user-level software (e.g., via a graphical interface). *Id.*; Ex-1002 ¶85.

## 2. Dependent claims 2-5 and 7

- a. **Claim 2: The processor of claim 1, wherein the first logic is to determine a portion of the power budget to allocate to the first domain based on the first sharing policy value.**

Finkelstein teaches Claim 2. Ex-1002 ¶¶87-89.

As discussed in Sections XII.A.1.e-f, Finkelstein discloses Claim 2. Moreover, as described above with respect to the “First Approach,” logic 140 implements power distribution, according to vector  $W_{ki}$ , wherein “entry  $i$  corresponds to the *portion of the budget that goes to power plane  $i$* ”:

$$E^{k,i} = W_{ki} E^k \quad (2)$$

Ex-1005 ¶34. Because the first domain corresponds to plane 1, entry 1,  $W_{k1}$ , corresponds to its portion of the power determined by logic 140. Ex-1002 ¶88.

- b. Claim 3: The processor of claim 1, wherein the first logic is to dynamically allocate the power budget further according to a first minimum reservation value for the first domain and a second minimum reservation value for the second domain, the first and second minimum reservation values controllable by the user-level software.**

Finkelstein teaches Claim 3. Ex-1002 ¶¶90-94.

Finkelstein discloses that the first logic is to dynamically allocate the power budget further according to first and second minimum reservation values for the first and second domains. Finkelstein's processors include logic 140 that dynamically allocates the power budget at run time. Ex-1005 ¶¶22, 24, 26; Section XII.A.1.e. Finkelstein also discloses that "a controller budget may be defined (e.g., implemented by the logic 140)" and provides several equations by which logic 140 may dynamically allocate the power budget.

$$f^i(E) = P_n^i, E \leq E_{low}^i$$

$$f^i(0) = P_1^1$$

$$f^i(E) = P_0^i, E \geq E_{high}^i$$

Ex-1005 ¶32. Finkelstein explains that  $f^i(E)$  is a "controller function" defined "[p]er power plane i" and is used to "map[] energy budget onto [a] discrete set of power states." *Id.* As shown in annotated Figure 4,  $f^i(E)$  "maps a range  $[E_{low}^i, E_{high}^i]$  onto



[a] discrete range of  $[P_n^i, P_0^i]$ ,” where  $P_n^i$  is the lowest power state. *Id.* Finkelstein also explains that “[b]udget values below  $E_{low}^i$  are mapped into  $P_n^i$ ,” confirming that  $P_n^i$  represents the minimum reservation value for power plane i. *Id.* Finkelstein therefore discloses a first minimum reservation value  $P_n^1$  for the first power plane and a second minimum reservation value  $P_n^2$  for the second power plane. Ex-1002 ¶92.

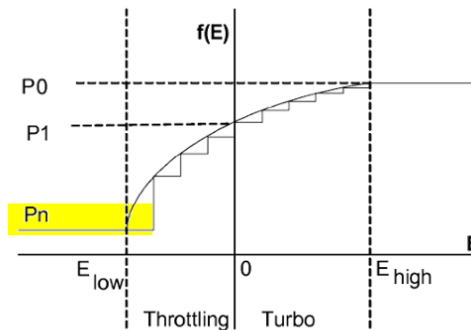


FIG. 4

Finkelstein also teaches that the first and second minimum reservation values are controllable by the user-level software. For example, Finkelstein discloses providing control values/constraints “by a software application or a user” and using “user (or application defined) preferences” to describe how to distribute a power budget among its power planes. Ex-1005 ¶¶29, 31, 34. A POSITA seeking to implement the power management logic of Finkelstein would have found it obvious that “a software application or a user” may control/provide the value of  $P_n^i$  because  $P_n^i$  is a constraint for power plane i and Finkelstein contemplates that (1) such a constraint may be provided “by a software application or a user,” and (2) a user may

define preferences to describe how to distribute a power budget among its power planes to satisfy  $P_n^i$  for each power plane. *Id.*; Ex-1002 ¶93.

- c. **Claim 4: The processor of claim 3, wherein the first logic is to provide at least a first portion of the power budget to the first domain, the first portion corresponding to the first minimum reservation value.**

Finkelstein teaches Claim 4. *See* Section XII.A.2.b; Ex-1002 ¶¶95-97. Specifically, Finkelstein discloses providing a first minimum reservation value,  $P_n^l$ , to the first power plane. *See* Ex-1005 ¶32.

- d. **Claim 5: The processor of claim 1, wherein the first logic is to determine the power budget for a current time interval based at least in part on a power budget carried forward from a previous time interval, a power consumed in the previous time interval, and a power budget decay value.**

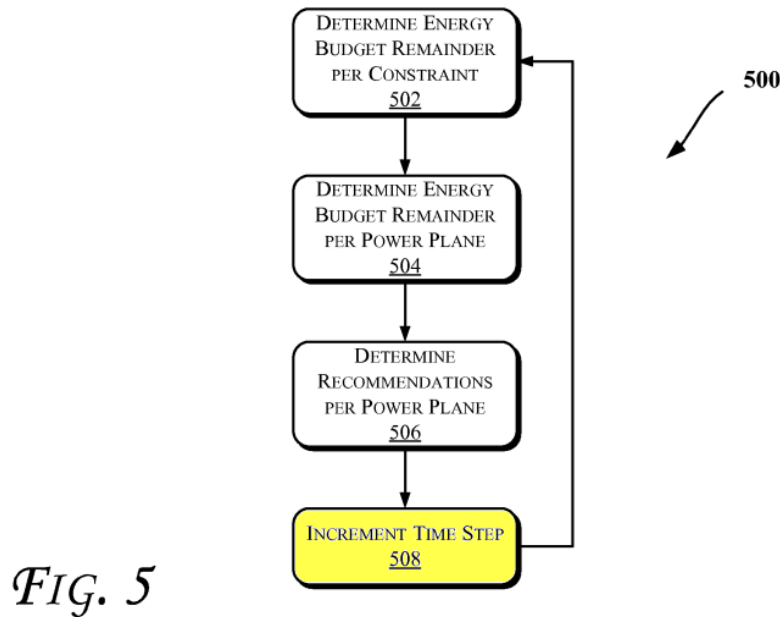
Finkelstein teaches Claim 5. Ex-1002 ¶¶98-101.

As discussed in Sections XII.A.1.e-f, specifically the “Second Approach,” Finkelstein discloses Claim 5. Finkelstein explains that “energy budget may be managed and/or power setting(s) (e.g., voltage and/or frequency changes) may be made *accordingly to a current budget*,” and that “[e]nergy-based power management may be performed by controlling the energy budget *defined iteratively* as:”

$$E_{n+1} = \alpha E_n + (TDP_n - P_n) \Delta t_n$$

where “ $TDP_n$  is the Thermal Design Power (TDP) power limit on step  $n$ ,” “ $P_n$  is the power spent on step  $n$  over time  $\Delta t_n$ ,” “ $\alpha$  is the decay component,” and “ $E_n$  corresponds to the energy ‘remainder’ which is the amount of energy not consumed by the system.” Ex-1005 ¶¶28-29.

A POSITA would have understood that  $E_n$  corresponds to a power budget carried forward from a previous time interval because  $E_n$  is “the energy ‘remainder’ which is the amount of energy not consumed by the system.” Ex-1005 ¶¶28-29; Ex-1002 ¶100. Similarly, a POSITA would have understood that  $P_n$  corresponds to a power consumed in the previous time interval because  $P_n$  is “power spent on step  $n$  over time  $\Delta t_n$ ” and that  $\alpha$  corresponds to a power budget decay value because  $\alpha$  is “the decay component.” Ex-1005 ¶¶28-29; Ex-1002 ¶100. Furthermore, as shown in Figure 5, Finkelstein discloses power management operations in which a new cycle begins with “determin[ing] energy budget remainder,” then “determin[ing] recommendations per power plane.” Ex-1005 ¶39. Hence, the power budget allocation in the current cycle is determined, at least in part, on the power budget carried forward from the last cycle. Ex-1002 ¶100.



- e. **Claim 7: The processor of claim 1, wherein the first logic is to dynamically allocate substantially all of the power budget to the first domain for a first workload, and to dynamically allocate substantially all of the power budget to the second domain for a second workload executed after the first workload.**

Finkelstein teaches Claim 7. Ex-1002 ¶¶102-105.

Finkelstein discloses that the first logic is to dynamically allocate substantially all of the power budget to the first domain for a first workload.<sup>4</sup> For example, Finkelstein discloses that “if multiple power planes sharing the same power source,” then these planes may “share a common package power/energy budget” so that, for example, “unused processor core power” of a core in one domain can be used for

---

<sup>4</sup> A POSITA would not have understood the full scope of “substantially all” with reasonable certainty. Regardless, “substantially all” would at least include “all of” or “the entire” budget as taught by Finkelstein.

cores in another domain that may be performing “intensive workload” in “Graphics Effect(s) (GFX).” Ex-1005 ¶26. Finkelstein also explains that in determining how to allocate the power budget to the power planes (domains), there are situations where “a single power plane may obtain *the entire budget*.” *Id.* ¶34; *see also id.* ¶¶26, 28. A POSITA would have understood that allocating the *entire* budget to one of the domains (e.g., a domain performing “intensive workload” in GFX) constitutes allocating “substantially all” of the power budget to that domain, notwithstanding that the lower-bound of “substantially all” may not be reasonably ascertainable in other contexts. Ex-1002 ¶103.

Moreover, a POSITA seeking to implement the teachings of Finkelstein would have found it obvious that Finkelstein’s power management logic 140 would dynamically allocate substantially all of the power budget to the second domain for a second workload executed after the first workload. Ex-1002 ¶104. A POSITA would have understood that processors encounter both graphic intensive workload and data processing intensive workload. *Id.* A POSITA would have also understood that processors may execute data processing intensive workload after execution of graphic intensive workload, and vice versa. *Id.* A POSITA would therefore have understood that after the core(s) in a first domain finishes performing “intensive workload” in GFX, another core in another domain may be re-allocated substantially all of the power budget (previously allocated to the first domain to perform

“intensive workload” in GFX) to perform data processing intensive workload. *Id.* Indeed, this would have been an obvious implementation step because Finkelstein already teaches to provide “*the entire budget*” to one power plane (domain). It thus would have been obvious to provide “*the entire budget*” to either domain, for essentially the same reasons and implemented in essentially the same way as taught by Finkelstein.

**B. Ground 2: Claim 7 is rendered obvious by Finkelstein in view of Conroy.**

**1. Dependent claim 7**

- a. Claim 7: The processor of claim 1, wherein the first logic is to dynamically allocate substantially all of the power budget to the first domain for a first workload, and to dynamically allocate substantially all of the power budget to the second domain for a second workload executed after the first workload.**

Finkelstein in view of Conroy teaches Claim 7. Ex-1002 ¶¶106-111.

As discussed in Section XII.A.2.e, Finkelstein teaches Claim 7. Moreover, a POSITA seeking to implement the power management logic of Finkelstein would also have found this feature obvious in view of Conroy. Ex-1002 ¶¶107-111.

Like Finkelstein, Conroy relates to “[m]ethods and apparatuses for dynamic power control.” Ex-1006, Title. Conroy also discloses a “power weighting arrangement between the subsystems,” and explains that “higher power draw is allocated to and allowed for the subsystem having a higher workload.” *Id.* ¶6.

Conroy further recognizes that many applications “present an *alternating* high workload to the CPU and to the GPU.” *Id.* ¶196.

A POSITA would have understood that Finkelstein’s processors may also execute applications that “present an alternating high workload to the CPU and to the GPU.” Ex-1006 ¶196; Ex-1002 ¶109. Thus, a POSITA would have been motivated to combine the teachings of Conroy with the teachings of Finkelstein and use Finkelstein’s processors to execute a “CPU-heavy load profile” after execution of a “GPU-heavy load profile,” and vice versa. Ex-1002 ¶109.

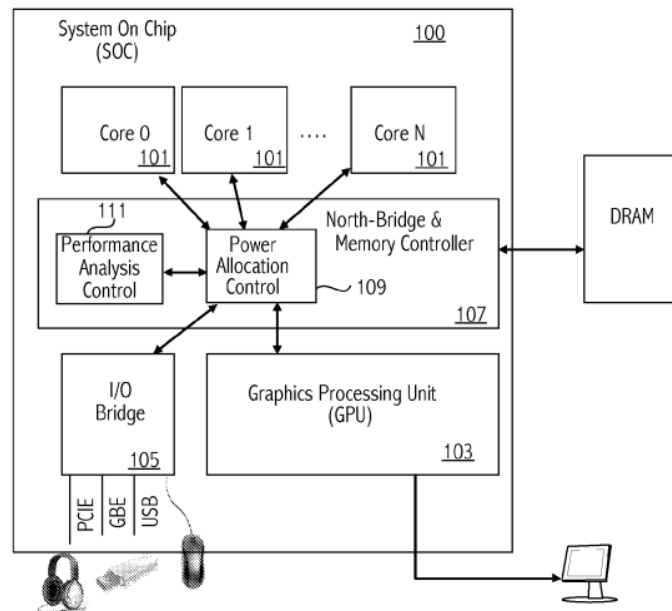
Additionally, a POSITA would have had a reasonable expectation of success in using Finkelstein’s processors to execute a “CPU-heavy load profile” after execution of a “GPU-heavy load profile,” and vice versa, because Finkelstein explicitly discloses that its power planes may “share a common package power/energy budget” so that, for example, “unused processor core power” of a first domain can be used for a second domain that may be performing “intensive workload” in “Graphics Effect(s) (GFX).” Ex-1005 ¶26; Ex-1002 ¶110. A POSITA would have understood that Finkelstein’s own power sharing logic can be utilized so that after the second domain performs an “intensive workload” in GFX (in other words, a “GPU-heavy load profile”), Finkelstein may allocate the “unused processor core power” to the first domain to perform a “CPU-heavy load profile,” and vice versa. Ex-1002 ¶110.

**C. Ground 3: Claims 13-15 and 17-18 are rendered obvious by Nussbaum.**

**1. Independent claim 13**

**a. Element 13[Pre]: A system comprising:**

Nussbaum teaches the preamble of Claim 13, to the extent it is limiting. Ex-1002 ¶¶112-114. For example, Nussbaum discloses a “System on a Chip (SOC) system” shown in Figure 1. Ex-1007 ¶¶10, 31, 43.



**FIG. 1**

**b. Element 13[A.i]: a multicore processor having a first domain including a plurality of cores, a second domain including a graphics engine, and**

Nussbaum teaches Element 13[A.i]. Ex-1002 ¶¶115-120.

Nussbaum teaches a multicore processor having a first domain including a plurality of cores. As shown in annotated Figure 1 below, Nussbaum’s SOC 100 “includes multiple CPU processing cores 101.” Ex-1007 ¶20.



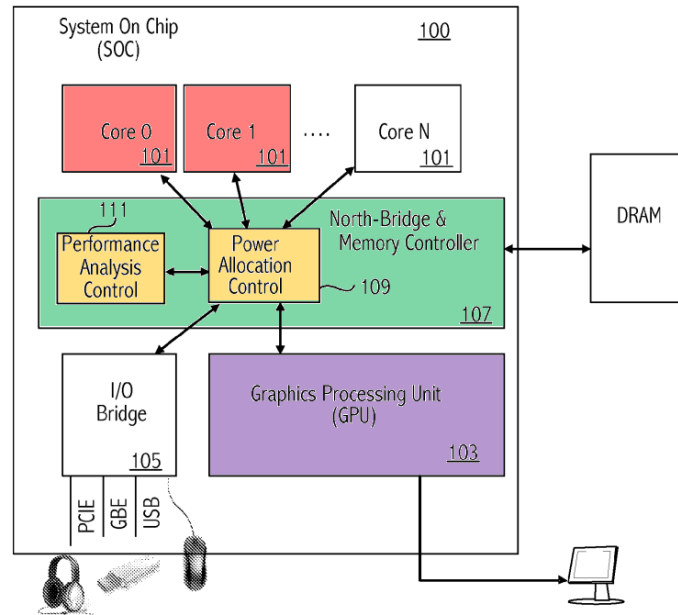


FIG. 1

Nussbaum also provides examples of power allocation in Tables 1 and 2, copied below, where a plurality of cores 101 (Core 0 and Core 1, red) are equally boosted to a “higher F, V to fill new power headroom.” Ex-1007 ¶¶21-24. A POSITA would have understood this to mean that Core 0 and Core 1 are operating in the same power domain. Ex-1002 ¶118. This understanding is consistent with the ’833 Patent’s description of a domain, because Core 0 and Core 1 form “a collection of hardware and/or logic that operates at the same voltage and frequency point.” Ex-1001, 1:58-60. Moreover, consistent with power allocations depicted in Tables 1 and 2 below, a POSITA seeking to implement the teachings of Nussbaum would also have found it obvious that multiple cores may operate in the same power domain (e.g., Core 0 and Core 1 may operate in a first domain while Core 2 and Core 3 may operate in another domain). Ex-1002 ¶118.

TABLE 1

On-die component	Allocated Power
Core0	8 w
Core1	8 w
Core2	8 w
Core3	8 w
GPU	5 w
Memory Controller	2 w
I/O Bridge	1 w
Total	40 w

TABLE 2

On-die component	Allocated Power	Remarks
Core0	16.75 w	Core can run at higher F, V to fill new power headroom
Core1	16.75 w	Core can run at higher F, V to fill new power headroom
Core2	0.5 w	Assume that idle core consumes 0.5 w
Core3	0.5 w	Assume that idle core consumes 0.5 w
GPU	2.5 w	
Memory Controller	2 w	
I/O Bridge	1 w	
Total	40 w	

Furthermore, Nussbaum discloses a second domain including a graphics engine. As shown in annotated Figure 1 and Tables 1 and 2 above, SOC 100 includes “GPU (Graphics Processing Unit) 103” (purple), which operates at different wattages compared to Core 0 and Core 1 (red). A POSITA would have understood this to mean that the GPU is not operating in the same domain as Core 0 and Core 1 (in other words, the GPU is operating in a second domain). Ex-1007 ¶¶20-24; Ex-1002 ¶119.

**c. Element 13[A.ii(1)]: a third domain including system agent circuitry,**

Nussbaum teaches Element 13[A.ii(1)]. Ex-1002 ¶¶121-125.

Nussbaum discloses system agent circuitry. As shown in annotated Figure 1, SOC 100 includes “North-Bridge & Memory Controller 107” (green), which includes “performance analysis control logic 111” and “power allocation controller 109” (yellow). Ex-1007 ¶20. Nussbaum teaches that “performance analysis control logic 111” and “power allocation controller 109” jointly serve as a system agent to “analyze[] performance sensitivity of the cores and other computational units” and “control[] allocation of the Thermal Design Point (TDP) power headroom to the on-die or on-platform components.” *Id.* ¶¶20, 39;<sup>5</sup> Ex-1002 ¶122.

---

<sup>5</sup> Consistent with the plain and ordinary meaning in the context of the '833 Patent, the specification refers to the component that controls the power supply to the cores and graphics, as the “system agent.” Ex-1002 ¶122 (citing, e.g., Ex-1001, 9:16-21).

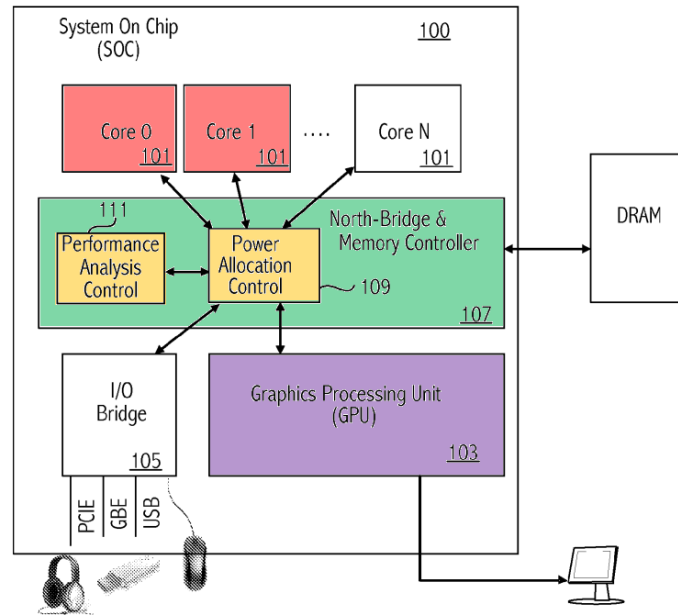


FIG. 1

Nussbaum also discloses that the system agent circuitry (performance analysis control logic 111 and power allocation controller 109, yellow) is included in North-Bridge & Memory Controller 107 (green) that operates in a third domain. Nussbaum provides an example of power reallocation in Tables 1 and 2, copied below, in which the power allocated to “Memory Controller” remains fixed at “2 w” despite changes made to other components (including Core 0 and Core 1 in the first domain and GPU in the second domain). Ex-1007 ¶¶21-24. Accordingly, a POSITA would have recognized that, because “Memory Controller” is combined with North-Bridge 107 and not called out separately in Tables 1 and 2, Nussbaum teaches that North-Bridge & Memory Controller 107 operates in a domain different from that of Core 0 and Core 1 (the first domain) and GPU (the second domain). Ex-1002 ¶124. In other

words, North-Bridge & Memory Controller 107, which includes the system agent circuitry, is operating in a third domain. *Id.*

TABLE 1

On-die component	Allocated Power
Core0	8 w
Core1	8 w
Core2	8 w
Core3	8 w
GPU	5 w
Memory Controller	2 w
I/O Bridge	1 w
Total	40 w

TABLE 2

On-die component	Allocated Power	Remarks
Core0	16.75 w	Core can run at higher F, V to fill new power headroom
Core1	16.75 w	Core can run at higher F, V to fill new power headroom
Core2	0.5 w	Assume that idle core consumes 0.5 w
Core3	0.5 w	Assume that idle core consumes 0.5 w
GPU	2.5 w	
Memory Controller	2 w	
I/O Bridge	1 w	
Total	40 w	

**d. Element 13[A.ii(2)]: the third domain to operate at a fixed power budget and**

Nussbaum teaches Element 13[A.ii(2)]. Ex-1002 ¶¶126-129.

As discussed in Section XII.C.1.c, Nussbaum discloses that the power allocated to North-Bridge & Memory Controller 107 remains fixed at “2 w” despite changes made to other components. Accordingly, a POSITA seeking to implement

the teachings of Nussbaum would have found it obvious to operate North-Bridge & Memory Controller 107 at a fixed power budget so that the power allocated to North-Bridge & Memory Controller 107 can remain fixed despite changes made to other components. Ex-1002 ¶¶127-128.

e. **Element 13[A.ii(3)]: including a power sharing logic to dynamically allocate a variable power budget between the first and second domains**

Nussbaum teaches Element 13[A.ii(3)]. Ex-1002 ¶¶130-133.

As shown in annotated Figure 1, Nussbaum explains that “performance analysis control logic 111” and “power allocation controller 109” (yellow) jointly serve as a system agent to “analyze[] performance sensitivity of the cores and other computational units” and “control[] allocation of the Thermal Design Point (TDP) *power headroom* to the on-die or on-platform components.” Ex-1007 ¶20; *see also id.* ¶¶19, 21-25, 39; Ex-1002 ¶¶122, 131. Nussbaum also discloses determining an “*available power headroom*” that can be dynamically allocated and explains that “[a]ny *change in the state* of the on-die components *triggers an update* of” this available power headroom, confirming that this power headroom is a variable. Ex-1007 ¶¶36, 39. Nussbaum further describes *selectively distributing* this power headroom *among* the computational units based on the computational units’ sensitivities to frequency changes. *Id.* ¶19; *see also id.* ¶¶20-25. Nussbaum also provides an example where “boost sensitivit[ies]” of two computational units are

used to “determine how to *allocate power between*” them. *Id.* ¶25. Moreover, Nussbaum explains that “[m]ost real-world applications are either CPU or GPU-bounded” and discloses “*reallocation of the power from idle or less active components to the busy components*” by having more power allocated to the busy ones.” *Id.* ¶¶22-23. Nussbaum further discloses “a workload sample” where GPU (purple, operating in a second domain) “operates at half power” so that certain cores (red, operating in a first domain) can be allocated more power. *Id.*; Section XII.C.1.b (explaining first and second domains). Accordingly, Nussbaum discloses a power sharing logic to dynamically allocate a variable power budget between the first and second domains. Ex-1002 ¶¶131-133.

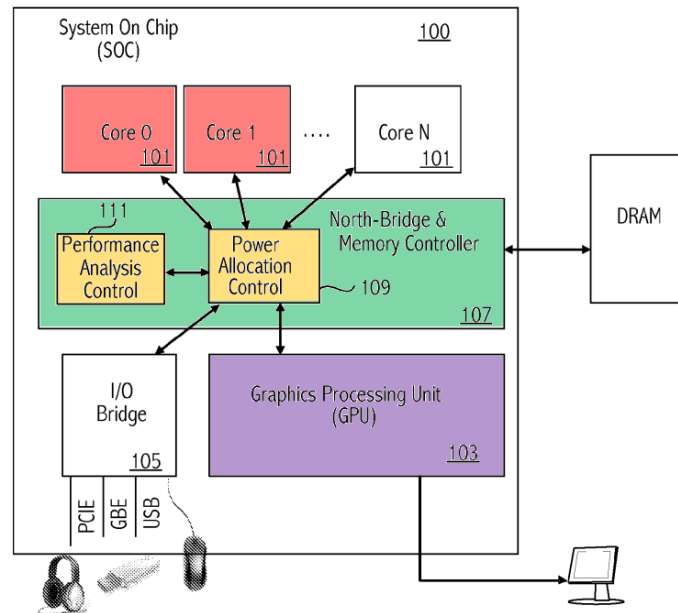


FIG. 1

- f. **Element 13[A.ii(4)]: based at least in part on a first power sharing value for the first domain stored in a first storage and a second power sharing value for the second domain stored in a second storage; and**

Nussbaum teaches Element 13[A.ii(4)]. Ex-1002 ¶¶134-137.

Nussbaum discloses that the power sharing logic allocates power between the first and second domains based at least in part on first and second power sharing values for the first and second domains. For example, Nussbaum describes *selectively distributing* a power headroom *among* its computational units based on the computational units' sensitivities to frequency changes, and provides an example where "boost sensitivit[ies]" of two computational units are used to "determine how to *allocate power between*" them. Ex-1007 ¶¶19, 25; *see also id.* ¶¶20-24. Nussbaum explains that the "boost sensitivity" indicates a "performance sensitivity of each computational unit to frequency change, and/or other change in performance capability," and further explains that the "computational units," for which boost sensitivities are determined, may include the cores operating in the first domain as well as the GPU operating in the second domain. *Id.* Accordingly, Nussbaum discloses allocating power between the first and second domains based at least in part on a boost sensitivity of a core operating in the first domain (i.e., a first power sharing value for the first domain) and a boost sensitivity of the GPU operating in the second domain (i.e., a second power sharing value for the second domain). Ex-1002 ¶135.



Nussbaum also teaches that first and second power sharing values are stored in first and second storages. For example, Nussbaum states that “boost sensitivity” is “determined and *stored on a computational unit basis*,” and describes a “boost sensitivity table” that stores boost sensitivity of each computational unit on a separate table entry. Ex-1007 ¶¶25, 32; *see also id.* ¶¶26-31, Fig. 4. Nussbaum further states that “the boost sensitivity table may be storage [sic] *within the SOC* 100,” and a POSITA seeking to implement the teachings of Nussbaum would have found it obvious to use registers located within the SOC to store the values contained in the boost sensitivity table so that the boost sensitivity table can be stored “within the SOC” and the values contained in the table can be stored on a “computational unit basis.” *Id.*; Ex-1002 ¶136. Indeed, Nussbaum itself discloses using registers of the SOC to store values associated with boost sensitivities. Ex-1007 ¶¶26-27, 30-31. Nussbaum therefore teaches that the first power sharing value for the first domain is stored in a first storage (corresponding to a first register that stores the boost sensitivity value for a core operating in the first domain) and the second power sharing value for the second domain is stored in a second storage (corresponding to a second register that stores the boost sensitivity value for the GPU operating in the second domain). Ex-1007 ¶¶25-32, Fig. 4; Ex-1002 ¶136.

**g. Element 13[B]: a dynamic random access memory (DRAM) coupled to the multicore processor.**

Nussbaum teaches Element 13[B]. Ex-1002 ¶¶138-140.

As shown in annotated Figure 1, Nussbaum discloses a DRAM (orange) coupled to the multicore processor. Ex-1007 ¶43.

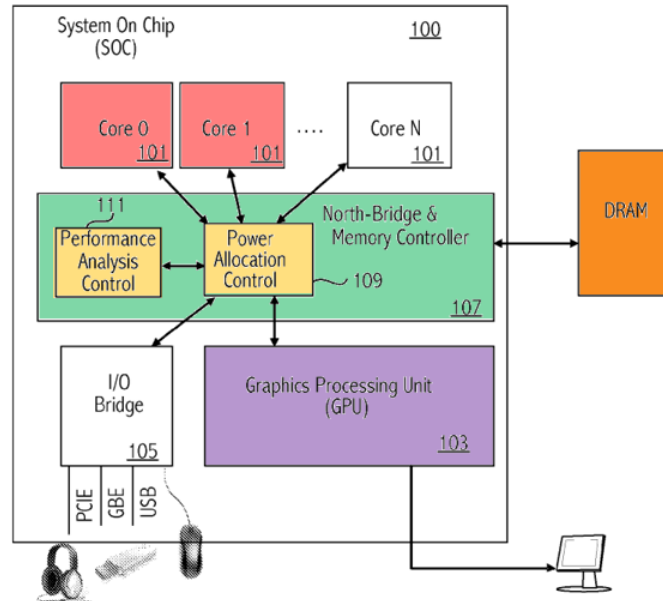


FIG. 1

## 2. Dependent claims 14-15 and 17-18

- a. **Claim 14:** The system of claim 13, wherein the power sharing logic to determine the variable power budget for a current time interval and to allocate a first portion of the variable power budget to the first domain according to the first power sharing value, and to allocate a second portion of the variable power budget to the second domain according to the second power sharing value.

Nussbaum teaches Claim 14. Ex-1002 ¶¶141-144.

Nussbaum discloses a power sharing logic to determine a variable power budget for a current time interval.<sup>6</sup> For example, Nussbaum discloses determining

---

<sup>6</sup> Petitioner reserves the rights to argue in the related district court or ITC matters

an “**available power headroom**” that can be dynamically allocated and explains that “[a]ny **change in the state** of the on-die components **triggers an update** of” this available power headroom. Ex-1007 ¶¶36, 39. Nussbaum further explains that “change of state that triggers the update” may include, e.g., “a change in performance or power state or change in application/workload activity,” “a process context change,” or “a performance state change,” confirming that the “available power headroom” is updated at run time for the current time interval. *Id.*; Ex-1002 ¶142.

And as discussed in Sections XII.C.1.e-f, Nussbaum also discloses allocating a first portion of the variable power budget (the available power headroom) to the first domain according to the first power sharing value, and allocating a second portion of the variable power budget to the second domain according to the second power sharing value.

- b. Claim 15: The system of claim 14, wherein the power sharing logic is to dynamically allocate substantially all of the variable power budget to the first domain for a first workload, and to dynamically allocate substantially all of the variable power budget to the second domain for a second workload executed after the first workload.**

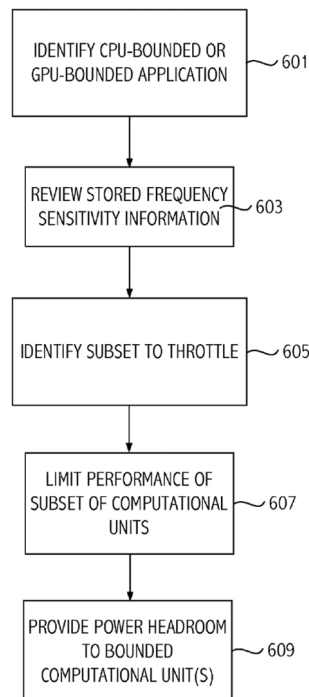
Nussbaum teaches Claim 15.<sup>7</sup> Ex-1002 ¶¶145-147.

---

that this claim limitation is indefinite because a POSITA would not understand the limitation’s bounds with respect to the phrase “power sharing logic to determine.”

<sup>7</sup> See *supra* note 4.

As shown in Figure 6, Nussbaum discloses providing “*the power headroom* made available through throttling” to “bounded computational unit(s).” Ex-1007 ¶47. Nussbaum also recognizes that “[m]ost real-world applications are either CPU or GPU-bounded.” Ex-1007 ¶¶22, 44, 46. Accordingly, Nussbaum explicitly contemplates providing the entire “*power headroom* made available through throttling” to CPU-bounded computational unit(s) when applications are CPU-bounded, including cores operating in the first domain. Ex-1002 ¶146. And when applications become GPU-bounded, the entire power headroom is made available to GPU-bounded computational unit(s), including GPU operating in the second domain. *Id.*



**FIG. 6**

- c. **Claim 17: The system of claim 14, wherein the power sharing logic is to further allocate the first portion of the variable power budget according to a first minimum reservation value for the first domain and allocate the second portion of the variable power budget according to a second minimum reservation value for the second domain.**

Nussbaum teaches Claim 17. Ex-1002 ¶¶148-151.

As shown in Table 3 below, Nussbaum discloses a “core power state,” referred to as “Deep Cstate,” in which cores operating in the first domain are allocated a minimum reservation value of Core\_DeepCState\_Pwr. Ex-1007 ¶33. Nussbaum also discloses “four performance states (P0, P1, P2, and P3)” for the cores operating in the first domain, and a POSITA would have understood that the operational point corresponding to the lowest performance state “P3” represents a minimum reservation value for the first domain when the cores are utilized to perform their operations. *Id.*; Ex-1002 ¶149.

TABLE 3

Core Performance States	Operational point (F, V)	Power (dynamic and static) consumed in this point	Remarks
P-boost	F-boost/V-boost	CoreBoostPwr	Boost point. Power budget of the Core has been exceeded
P0	F0/V0	Core_Pwr0	Core Power Budget
P1	F1/V1	Core_Pwr1	
P2	F2/V2	Core_Pwr2	
P3	F3/V3	Core_Pwr3	
Idle	Clocks Off/Low voltage	Core_Idle_Pwr	
Deep Cstate	Clocks Off/Power Off	Core_DeepCstate_Pwr	Core is either power gated or deep voltage is applied

Similarly, as shown in Table 4 below, Nussbaum discloses “four power states” for the GPU operating in the second domain, and a POSITA would have understood that the operational point corresponding to “GPU\_P3” represents a second minimum reservation value for the second domain because it corresponds to the lower performance state the GPU in the second domain may be utilized to perform its operations. Ex-1007 ¶¶34-38; Ex-1002 ¶158.

TABLE 4

GPU Performance States	GPU Power (dynamic and static) consumed in this point
GPU-boost	GPUBoostPwr
GPU_P0	GPU_Pwr0
GPU_P1	GPU_Pwr1
GPU_P2	GPU_Pwr2
GPU_P3	GPU_Pwr3

- d. **Claim 18: The system of claim 14, wherein the power sharing logic is to further allocate the variable power budget according to a preference value, the preference value to favor the second domain over the first domain.**

Nussbaum teaches Claim 18. Ex-1002 ¶¶152-154.

As shown in Figure 6, Nussbaum discloses allocating its variable power budget based on an output value of step 601 that identifies whether an application is “CPU-bounded” or “GPU-bounded.” Ex-1007 ¶¶44-48. Nussbaum explains that if the output value of step 601 identifies an application as “GPU-bounded,” then “CPU cores may be throttled” to “release power margin available to the GPU.” *Id.* ¶44. Thus, a POSITA would have understood that an output value of step 601 being an

identification of a “GPU-bounded” application is a preference value to favor the second domain (which includes the GPU) over the first domain (which includes the cores). Ex-1002 ¶153.

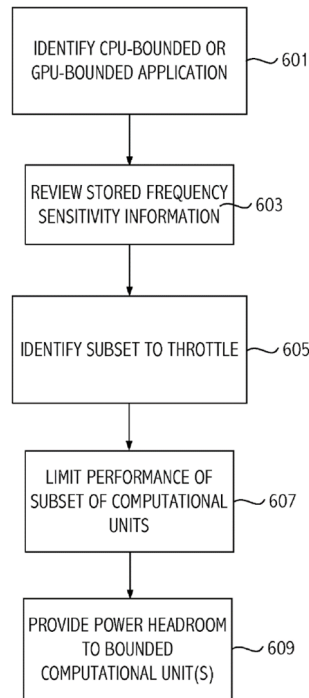


FIG. 6

**D. Ground 4: Claim 16 is rendered obvious by Nussbaum in view of Bose.**

**1. Dependent claim 16**

- a. Claim 16: The system of claim 14, wherein the power sharing logic is to increment the first power sharing value when a request for a higher frequency for the first domain is not granted, and to increment the second power sharing value when a request for a higher frequency for the second domain is not granted.**

Nussbaum in view of Bose teaches Claim 16. Ex-1002 ¶¶155-159.

Nussbaum discloses “determin[ing] **boost sensitivity** of a computational unit” and explains that the boost sensitivity of a core “represents the sensitivity of the core, running that particular process context, to a change in frequency.” Ex-1007 ¶¶7, 27, 32. Nussbaum further discloses that “[t]he boost sensitivity for each core is tied to the current processor context, [and] when the context changes, the **sensitivity is re-evaluated.**” *Id.* ¶30. Nussbaum also explains that boost sensitivity may “expire[] for each context based on a fixed or programmable timer,” and in some implementations, “both a timer and context switch, whichever occurs first, are used to initiate the boost sensitivity reevaluation.” *Id.* Thus, a POSITA would have understood that Nussbaum explicitly contemplates changing (e.g., incrementing) boost sensitivities of computational units, including computational units whose requests for a higher frequency were not granted. Ex-1002 ¶156. A POSITA would have also understood that boost sensitivity reevaluation is applicable equally to the first and second domains, rendering it obvious for Nussbaum to increment the first power sharing value when a request for a higher frequency for the first domain is not granted, and to increment the second power sharing value when a request for a higher frequency for the second domain is not granted. *Id.*

Moreover, a POSITA seeking to implement the teachings of Nussbaum would also have found this feature obvious in view of Bose. Ex-1002 ¶¶157-158. Like Nussbaum, Bose discloses “power management of a system of connected



components ... wherein each component is assigned a power budget as determined by a number of allocated tokens in the token allocation map.” Ex-1008, Abstract. Bose utilizes a “token management unit (TMU)” to manage allocation of tokens to “token-managed units,” such as “processor resources,” and explains that “[e]ach managed component, in response to its own operating conditions ... can either attempt to donate one or more token(s)” or “read (receive) one or more token(s),” e.g., by “request[ing] for token read/write operations ... to be serviced.” *Id.* ¶¶38-39, 68. Bose further discloses that if the performance level of a particular resource is “below a predefined minimum performance,” a process is initiated to “***try and read additional tokens*** from [a] centralized bus token storage ... to boost up performance” of that particular resource, e.g., “by ***increasing its voltage and frequency***, if allowed by the system architecture.” *Id.* ¶86. And as shown in annotated Figure 13 below, Bose explains that if no additional tokens can be allocated to the particular resource at the moment (i.e., a request for a higher frequency is not granted), that particular resource can wait for its token count to be increased in the next iteration of the process. *Id.* ¶86, Fig. 13.

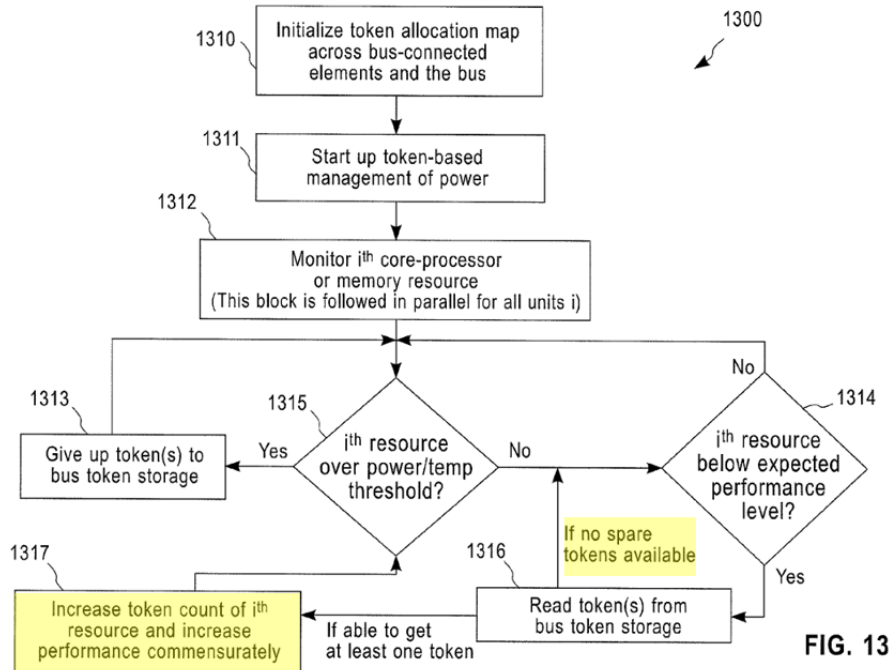


FIG. 13

A POSITA would have been motivated to implement Nussbaum's power allocation operations with Bose's teaching to increment a boost sensitivity of a particular domain (e.g., the first or the second domain) when a request for a higher frequency for that particular domain is not granted, especially if that particular domain is operating below an expected performance level, so as to avoid continued denial of that particular domain's request for a higher frequency and to avoid starvation of that domain. Ex-1002 ¶158. A POSITA would have had a reasonable expectation of success because Nussbaum explicitly contemplates re-evaluating its boost sensitivities. Ex-1007 ¶30; Ex-1002 ¶158.

**E. Ground 5: Claims 13-15 and 17-18 are rendered obvious by Conroy.**

**1. Independent claim 13**

**a. Element 13[Pre]**

Conroy teaches the preamble of Claim 13, to the extent it is limiting. Ex-1002 ¶¶160-162. For example, Conroy discloses “methods and apparatuses to dynamically redistribute power in a *system* that includes a plurality of subsystems.” Ex-1006, Abstract.

**b. Element 13[A.i]: a multicore processor having a first domain including a plurality of cores, a second domain including a graphics engine, and**

Conroy teaches Element 13[A.i]. Ex-1002 ¶¶163-168.

Conroy teaches a multicore processor having a first domain including a plurality of cores. As shown in annotated Figure 37 below, Conroy discloses “system 3700,” which a POSITA would have understood to be a processor. Ex-1002 ¶164. Moreover, to the extent it is argued that a processor must be implemented on a chip, a POSITA seeking to implement the teachings of Conroy would have found that to be obvious because doing so would improve the reliability and power efficiency of system 3700 while also reducing its size and weight. *Id.* Conroy states that system 3700 includes “subsystem A (e.g., CPU)” 3701 (red), and that “[c]omponents of the system 3700, including *processors* [and] *microcontrollers*,... are described in detail ... with respect to FIGS. 1-17.” Ex-1006 ¶225. One of these figures, namely

Figure 12 (annotated below), depicts a CPU (red) as a plurality of “main microprocessor(s)” 1201. *Id.* ¶167.

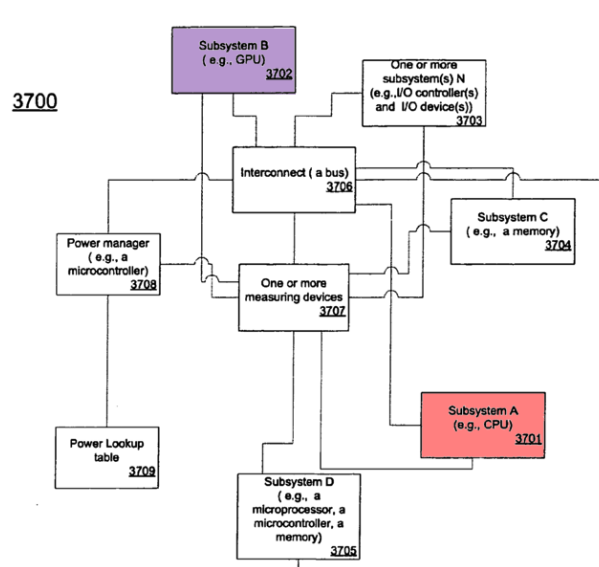


FIG. 37

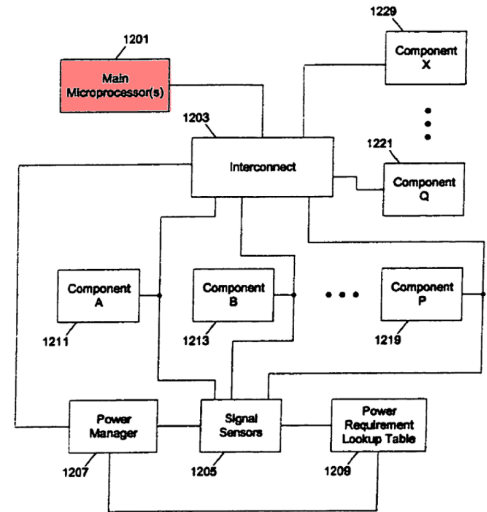
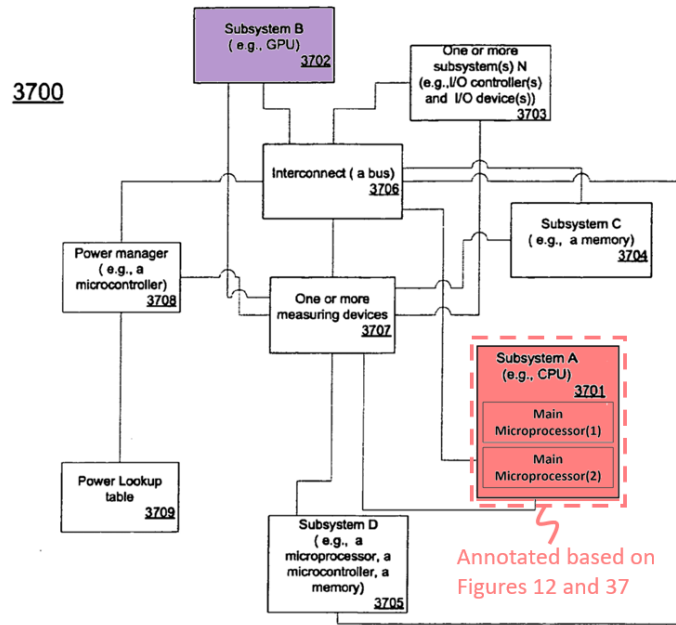


Fig. 12

Thus, a POSITA would have understood that Conroy explicitly contemplates its Subsystem A 3701 (red) to include a plurality of “main microprocessor(s),” which correspond to a plurality of cores, as shown in the composite Figure 37' below. Ex-1006 ¶¶50, 65-66, 167, 188-194, 225, Figs. 12, 37; Ex-1002 ¶166. Furthermore, Conroy discloses using “power manager” to determine “*a throttle setting*” for the plurality of “main *microprocessor(s)*,” confirming that Conroy explicitly contemplates having a plurality of cores operating in the same domain with the same “throttle setting.” Ex-1006 ¶¶167, 225; Ex-1002 ¶166.



Composite Figure 37'

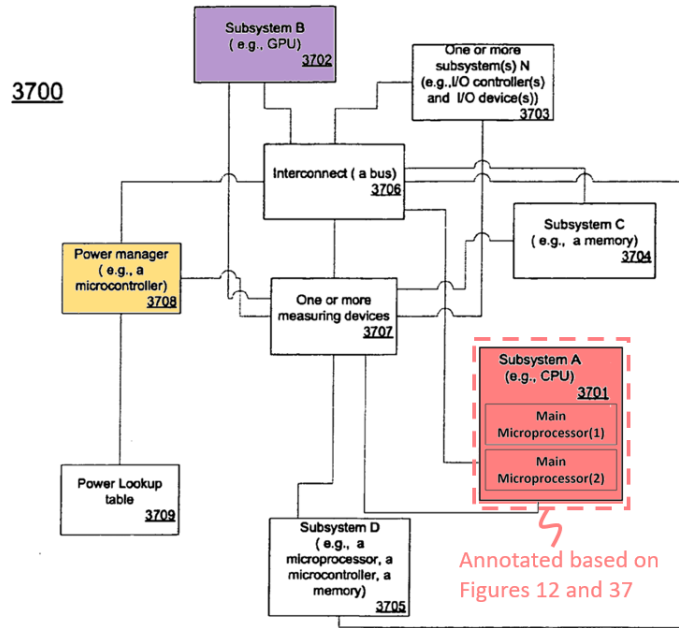
Conroy also discloses a second domain including a graphics engine. As shown in annotated Figure 37, system 3700 includes “subsystem B (e.g., GPU)” 3702 (purple). Ex-1006 ¶225. Conroy explains that during a “CPU-heavy load profile,” power may be distributed such that “CPU consumes about 75%, while GPU consumes about 25% of the power allocated for the system,” and vice versa. *Id.* ¶¶196-197, 203. Thus, Conroy’s GPU (purple) operates in a second domain. *Id.*; Ex-1002 ¶167.

**c. Element 13[A.ii(1)]: a third domain including system agent circuitry,**

Conroy teaches Element 13[A.ii(1)]. Ex-1002 ¶¶169-171.

Conroy discloses that system 3700 includes “power manager 3708.” Ex-1006 ¶225. Conroy explains that its power manager (e.g., a microcontroller) is “used to

budget the power usage dynamically” and “can be considered as part of the non-throttled components.” *Id.* ¶¶94, 168. Thus, Conroy teaches a third (non-throttled) domain including system agent circuitry (power manager, yellow). *Id.* ¶¶94, 168, 225, Figs. 4, 12, 37; Ex-1002 ¶170; *see also supra* note 5.



Composite Figure 37'

**d. Element 13[A.ii(2)]: the third domain to operate at a fixed power budget and**

Conroy teaches Element 13[A.ii(2)]. Ex-1002 ¶¶172-174.

Conroy states that its power manager “used to budget the power usage dynamically ... can be considered as part of the *non-throttled* components,” which are “not throttled to trade performance for power usage.” Ex-1006 ¶¶81, 94. Conroy also discloses determining a power budget for the non-throttled components by “adding together the maximum powers which could be consumed” by the non-

throttled components, and explains that this calculation “can be done when the system is designed; and the result can be a *constant*.” *Id.* ¶150. Conroy therefore teaches allocating a fixed power budget (a *constant* that is sufficient to accommodate the sum of “the maximum powers which could be consumed by the non-throttled” components) to the non-throttled components. Ex-1002 ¶173.

**e. Element 13[A.ii(3)]: including a power sharing logic to dynamically allocate a variable power budget between the first and second domains**

Conroy teaches Element 13[A.ii(3)]. Ex-1002 ¶¶175-177.

Conroy discloses “power manager (e.g., a microcontroller)” 3708 and uses this microcontroller to “execut[e] sequences of instructions contained in a memory” to “dynamically budget power usage and determine throttle settings.” Ex-1006 ¶¶190-191, 194, 225; *see also id.* ¶¶80-82, Fig. 3 (“[D]ynamically determined power budget (301) is to be allocated to different components (subsystems) of the system.”).

**f. Element 13[A.ii(4)]: based at least in part on a first power sharing value for the first domain stored in a first storage and a second power sharing value for the second domain stored in a second storage; and**

Conroy teaches Element 13[A.ii(4)]. Ex-1002 ¶¶178-182.

Conroy discloses dynamically allocating a power budget according to first and second sharing policy values for the first and second domains. For example, Conroy discloses implementations in which “a workload of a subsystem determines the

amount of power used by the subsystem” and explains that the power may be “redistributed in an asymmetric fashion, tracking the workloads of each of the subsystems.” Ex-1006 ¶¶196-197. During a “CPU-heavy load profile,” when the workload in the first domain is intensive, power may be distributed such that “CPU consumes about 75%, while GPU consumes about 25% of the power allocated for the system,” and vice versa. *Id.* ¶203. Conroy further discloses selecting a “***power weighting arrangement among the subsystems*** ... based on the load profile” and states that “the power weighting arrangement ... may be a power distribution table ***selected out of a plurality of power distribution tables.***” *Id.* ¶¶6, 200. Examples of power distribution tables are illustrated in Figures 21A-C and 22, copied below, confirming that Conroy discloses dynamically allocating a power budget according to first and second sharing policy values (power weighting arrangement settings selected from a power distribution table based on load profile) for the first and second domains. *Id.* ¶¶6, 200, Figs. 21A-C, 22.



Balanced load profile K1 (e.g., 0.5, 0.5)

2100

Settings	CPU	GPU	Allocated Power
Setting 1	A1 (e.g., 10)	B1 (e.g., 10)	P1 (e.g., 20)
Setting 2	A2 (e.g., 7)	B2 (e.g., 7)	P2 (e.g., 14)
Setting 3	A3 (e.g., 5)	B3 (e.g., 5)	P3 (e.g., 10)

Fig. 21 A

2110

CPU-heavy load profile K2 (e.g., 0.75, 0.25)

Settings	CPU	GPU	Allocated Power
Setting 1	A1 (e.g., 15)	B1 (e.g., 5)	P1 (e.g., 20)
Setting 2	A2 (e.g., 10)	B2 (e.g., 4)	P2 (e.g., 14)
Setting 3	A3 (e.g., 7)	B3 (e.g., 3)	P3 (e.g., 10)

Fig. 21 B

2120

GPU-heavy load profile K3 (e.g., 0.25, 0.75)

Settings	CPU	GPU	Allocated Power
Setting 1	A1 (e.g., 5)	B1 (e.g., 15)	P1 (e.g., 20)
Setting 2	A2 (e.g., 3.5)	B2 (e.g., 10)	P2 (e.g., 14)
Setting 3	A3 (e.g., 3)	B3 (e.g., 7)	P3 (e.g., 10)

Fig. 21 C

Load Profile Kn

2200

Settings	Subsystem 1	Subsystem 2	Subsystem...	Subsystem N	Power
1	A1	B1	...	C1	P1
2	A2	B2	...	C2	P2
M	Am	Bm	...	Cm	Pm

Fig. 22

Conroy also teaches that first and second power sharing values are stored in first and second storage. For example, Conroy explains that its “power distribution table” is described “with respect to FIGS. 7, 21, and 22.” Ex-1006 ¶225. Conroy’s Figure 7, copied below, shows a power distribution table that stores power settings for various system components, and Conroy describes an example where these

power settings are stored in registers. *Id.* ¶97 (“For example, the microcontroller may store [a] determined throttle setting in a register.”).

701

Increasing in  
a Target Goal  
Level

↓

Global Throttle Setting 1	Setting 1 for Component A	...	Setting 1 for Component X	Power Usage Requirement 1
Global Throttle Setting 2	Setting 1 for Component A	...	Setting 1 for Component X	Power Usage Requirement 2
Global Throttle Setting 3	Setting 2 for Component A	...	Setting 1 for Component X	Power Usage Requirement 3
⋮				
Global Throttle Setting n	Setting m for Component A	...	Setting q for Component X	Power Usage Requirement N

Fig. 7

Similarly, Figures 21 and 22, above, also depict power distribution tables that store power settings for various system components. Thus, a POSITA seeking to implement the teachings of Conroy would have found it obvious to use registers to store the various power settings depicted in Conroy’s Figures 7, 21, and 22. *Id.* ¶¶97, 225; Ex-1002 ¶181. Accordingly, Conroy teaches storing a first power sharing value for a first domain in a first storage (corresponding to a first register that stores a power setting for a CPU operating in a first domain) and storing a second power sharing value for a second domain in a second storage (corresponding to a second

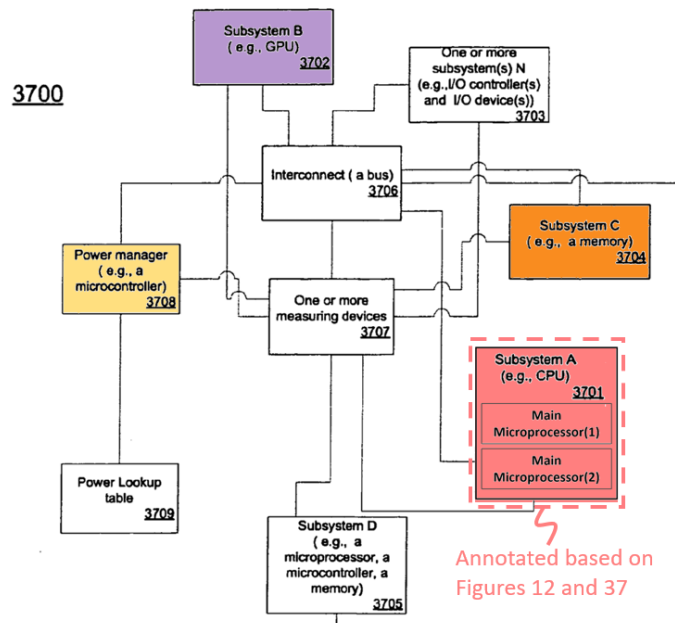
register that stores a power setting for a GPU operating in a second domain). *Id.*

¶¶97, 225, Figs. 7, 21, 22; Ex-1002 ¶¶179-182.

**g. Element 13[B]: a dynamic random access memory (DRAM) coupled to the multicore processor.**

Conroy teaches Element 13[B]. Ex-1002 ¶¶183-185.

As shown in composite Figure 37', Conroy discloses “subsystem C” 3704 (orange), which “may be a volatile RAM.” Ex-1006 ¶225, Fig. 37. Conroy also explains that “volatile RAM” is “typically implemented as dynamic RAM (DRAM).” *Id.* ¶187.



Composite Figure 37'

2. **Dependent claims 14-15 and 17-18**

- a. **Claim 14: The system of claim 13, wherein the power sharing logic to determine the variable power budget for a current time interval and to allocate a first portion of the variable power budget to the first domain according to the first power sharing value, and to allocate a second portion of the variable power budget to the second domain according to the second power sharing value.**

Conroy teaches Claim 14. Ex-1002 ¶¶186-189.

Conroy discloses a power sharing logic to determine a variable power budget for a current time interval. For example, Conroy uses “the *past power usage* (e.g., measurements ... at times  $T-(n-1)\Delta$ ,  $T-(n-2)\Delta$ , ... ,  $T$ )” to “*dynamically determine the allowable power budget*” for time interval  $T+\Delta$ , and allocates this “dynamically determined” power budget to different components of the system. Ex-1006 ¶¶12, 70-72, 80-83, 90, 120. A POSITA would have understood time interval  $T+\Delta$  to be a “current time interval” because Conroy refers to the time intervals preceding  $T+\Delta$  as being in the “*past*.” *Id.* ¶¶70-72, Fig. 2; Ex-1002 ¶187.

And as discussed in Sections XII.E.1.e-f, Conroy also discloses allocating a first portion of the variable power budget (the dynamically determined “allowable power budget”) to the first domain according to the first power sharing value, and allocating a second portion of the variable power budget to the second domain according to the second power sharing value.

- b. **Claim 15: The system of claim 14, wherein the power sharing logic is to dynamically allocate substantially all of the variable power budget to the first domain for a first workload, and to dynamically allocate substantially all of the variable power budget to the second domain for a second workload executed after the first workload.**

Conroy teaches Claim 15.<sup>8</sup> Ex-1002 ¶¶190-192.

Conroy discloses implementations in which “a workload of a subsystem determines the amount of power used by the subsystem” and recognizes that “program development tools and scientific applications present a *high load to the CPU*, but *almost no load to the GPU*.” Ex-1006 ¶196. Conroy also recognizes that many applications “present an alternating high workload to the CPU and to the GPU that results in an *alternating asymmetric load profile* of the system.” *Id.* Conroy explains that for a “CPU-heavy load profile,” CPU may “consume[] a *substantially bigger* share of the total system’s power, while a graphics processor is *hardly used at all*,” and that for a “GPU-heavy load profile,” GPU may “consume[] a *substantially bigger* share of the total system’s power, while a CPU is *hardly used at all*.” *Id.* ¶203. Thus, Conroy explicitly contemplates redistributing substantially all of its variable power budget to the CPU when the GPU has “almost no load” or is “hardly used at all,” and vice versa. Ex-1002 ¶191.

---

<sup>8</sup> See *supra* note 4.

- c. **Claim 17: The system of claim 14, wherein the power sharing logic is to further allocate the first portion of the variable power budget according to a first minimum reservation value for the first domain and allocate the second portion of the variable power budget according to a second minimum reservation value for the second domain.**

Conroy teaches Claim 17. Ex-1002 ¶¶193-195.

Conroy discloses a “power usage range” between a “minimum possible power consumption” and a “maximum power consumption.” Ex-1006 ¶53, Fig. 1. Conroy also explains that even when a subsystem has the lowest power allocation utilization of 0, Conroy’s power manager still allocates power to that subsystem to maintain the subsystem in a “low power mode.” *Id.* ¶198. Conroy further discloses sensing, predicting, or estimating “the power used by *each of* the subsystems,” confirming that the “minimum possible power consumption” represents an allocation that, at a minimum, allocates portions of the variable power budget to satisfy the minimum reservation values for each of the subsystems operating in the first and second domains. *Id.* ¶¶6, 8, 198, 202-206, Abstract; Ex-1002 ¶194.

- d. **Claim 18: The system of claim 14, wherein the power sharing logic is to further allocate the variable power budget according to a preference value, the preference value to favor the second domain over the first domain.**

Conroy teaches Claim 18. Ex-1002 ¶¶196-198.

Conroy discloses allocating the variable power budget based on whether a load profile is CPU or GPU-heavy. Ex-1006 ¶¶196, 203. Conroy also explains that

a determination of a load profile being “GPU-heavy” means the GPU may “consume[] a substantially bigger share of the total system’s power, while a CPU is hardly used at all.” *Id.* Conroy further provides an example of such a “GPU-heavy” load profile, referred to as “K3 e.g., [0.25, 0.75].” *Id.* Thus, a POSITA would have understood that a “GPU-heavy” load profile, such as K3, represents a preference value to favor the second domain (which includes the GPU) over the first domain (which includes the CPU). Ex-1002 ¶197.

**F. Ground 6: Claim 16 is rendered obvious by Conroy in view of Bose.**

**1. Dependent claim 16**

- a. Claim 16: The system of claim 14, wherein the power sharing logic is to increment the first power sharing value when a request for a higher frequency for the first domain is not granted, and to increment the second power sharing value when a request for a higher frequency for the second domain is not granted.**

Conroy in view of Bose teaches Claim 16. Ex-1002 ¶¶199-203.

Conroy discloses incrementing the first and second power sharing values for the first and second domains. As shown in Figure 22, Conroy recognizes that system performance “may move up and down across column 2206,” and “if the performance of the system *requires an increase* in total power, the performance *moves up* across the column e.g., from system’s setting M to system’s setting 1.” Ex-1006 ¶204.

Load Profile Kn

2200

2208 {	2201	2202	2203	2204	2205	2206
	Settings	Subsystem 1	Subsystem 2	Subsystem...	Subsystem N	Power
	1	A1	B1	...	C1	P1
	2	A2	B2	...	C2	P2
	M	Am	Bm	...	Cm	Pm

Fig. 22

Conroy also recognizes that “[t]ypically, a selected throttle setting is used *until the power measurement for the next time interval is obtained and the next iteration of dynamic throttling is performed.*” Ex-1006 ¶130. Thus, a POSITA would have understood that Conroy explicitly contemplates not granting a subsystem’s request to increase the system’s setting immediately; instead, Conroy teaches waiting till the next iteration to “move[] up” its system’s setting. Ex-1002 ¶201.

Moreover, a POSITA seeking to implement the teachings of Conroy would also have found this feature obvious in view of Bose, which discloses instructing a component to wait for its token count to be increased in the next iteration of the process if its request for a higher frequency is not granted. Section XII.D.1.a; Ex-1002 ¶202. A POSITA would have been motivated to implement Conroy’s power manager with Bose’s teaching to not grant a subsystem’s request to increase the system’s setting immediately and, instead, wait till the next iteration to do so in



order to keep the power supplied to the various subsystems stable within each integration. Ex-1002 ¶202. A POSITA would have had a reasonable expectation of success in doing so given that Conroy itself explicitly contemplates such implementations. Ex-1006 ¶130; Ex-1002 ¶202.

**G. Ground 7: Claims 1-5, 7, 13-15, and 17-18 are rendered obvious by Conroy in view of Finkelstein.**

**1. A POSITA would be motivated to combine the teachings of Conroy and Finkelstein, and would have a reasonable expectation of success in doing so.**

In addition to the reasons set out below with respect to specific claim elements, a POSITA would have been motivated to combine the teachings of Conroy and Finkelstein, and would have had a reasonable expectation of success in doing so, because each relates to the same well-known issues and describe similar methods for managing power allocation in a system with multiple cores or subsystems.

Conroy discloses “dynamically redistribut[ing] power in a system that includes a plurality of subsystems.” Ex-1006, Abstract. For example, Conroy teaches identifying a “load profile of the system” and redistributing power “between the subsystems while tracking the load profile.” *Id.* Conroy also discloses a “power weighting arrangement between the subsystems,” and explains that “higher power draw is allocated to and allowed for the subsystem having a higher workload.” *Id.* ¶6.

Like Conroy, Finkelstein discloses distributing a power budget and provides several formulae by which a power management logic may dynamically allocate the

power budget between multiple power planes. Ex-1005 ¶¶28-36. And, again like Conroy, Finkelstein also discloses sharing an “energy budget” so that “unused processor core power” can be used for other cores that may be performing more “intensive workload.” *Id.* ¶26.

While Conroy discloses dynamically redistributing power in a data processing and computer systems generally, Finkelstein discloses implementing such a system on one particular type of such a system—*i.e.*, an integrated circuit (IC) chip. *See, e.g.*, Ex-1005 ¶¶2, 11, 46; Ex-1006, Abstract, ¶225, Fig. 37.

A POSITA would have been motivated to combine the teachings of Conroy and Finkelstein for each of several independent reasons. Ex-1002 ¶208. First, a POSITA would have been motivated to implement Conroy’s system on a chip, as in Finkelstein, because a POSITA would have recognized that integrated circuits such as Finkelstein’s allowed for the desirable “integrat[ion] of additional functionality onto a single silicon substrate” (Ex-1005 ¶2), which a POSITA would have expected to reduce the size and weight of a system with a given set of functionality. *Id.* Second, a POSITA would have been motivated to implement Conroy’s system on a chip, as in Finkelstein, because doing so would have been expected to improve the reliability and power efficiency of the system. *Id.* Additionally, a POSITA would have had a reasonable expectation of success in doing so, at least because (1) each of these solutions involves routine hardware and software functionalities that are

amenable to predictable combination by a POSITA, and (2) such implementations have already been provided in Finkelstein. *Id.*

## 2. Independent claim 1

### a. Element 1[Pre]: A processor comprising:

Conroy alone or in view of Finkelstein teaches the preamble of Claim 1, to the extent it is limiting. Ex-1002 ¶¶209-212.

As shown in Figure 37, Conroy discloses “system 3700” that includes a CPU, a GPU, and a power manager. Ex-1006 ¶225. A POSITA would have understood such a system to be a processor. *See* Section XII.E.1.b; Ex-1002 ¶¶164, 210.

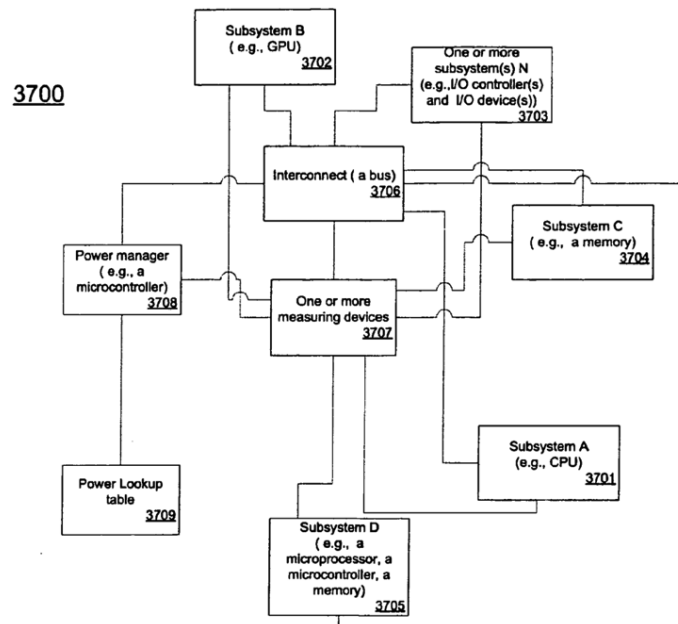


FIG. 37

Moreover, a POSITA would also have found it obvious to implement system 3700 on a chip in view of Finkelstein. Ex-1002 ¶211. Like Conroy,

Finkelstein discloses a system that includes a CPU, a graphics circuit, and a power manager. Ex-1005 ¶¶11, 42-43, 48-50, Figs. 1, 6, 7. Finkelstein states that its components “may be implemented on a single integrated circuit (IC) chip” and refers to such a single IC chip as a “processor.” *Id.* A POSITA seeking to implement the teachings of Conroy would have been motivated to implement Conroy’s system 3700 on a single IC chip to improve its reliability and power efficiency while also reducing its size and weight. Ex-1002 ¶¶164, 211. Additionally, a POSITA would have had a reasonable expectation of success in doing so because such implementations have already been provided in Finkelstein. Ex-1002 ¶211.

**b. Element 1[A]: a first domain and a second domain, each of the first and second domains to operate at an independent voltage and frequency;**

Conroy teaches Element 1[A]. Ex-1002 ¶¶213-218.

Conroy teaches a first domain and a second domain. As shown in annotated Figure 37 below, Conroy discloses “system 3700” that includes “subsystem A (e.g., CPU)” 3701 (red) and “subsystem B (e.g., GPU)” 3702 (purple). Conroy explains that during a “CPU-heavy load profile,” power may be distributed such that “CPU consumes about 75%, while GPU consumes about 25% of the power allocated for the system,” and vice versa. Ex-1006 ¶¶196-197, 203. Thus, Conroy’s CPU (red) and GPU (purple) operate in different power domains. *Id.*; Ex-1002 ¶214.

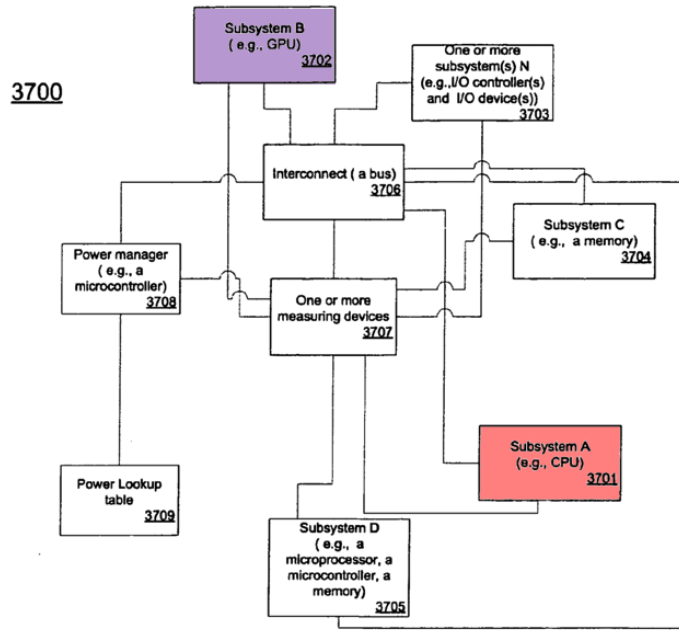


FIG. 37

Conroy also discloses that each of the first and second domains operates at an independent voltage and frequency. For example, Conroy discloses a “power weighting arrangement,” which “may be a power distribution table selected out of a plurality of power distribution tables” that describe, among other things, “power used by each of the subsystems at various system’s settings.” Ex-1006 ¶¶5-6. Conroy explains that its “power distribution table” is described “with respect to FIGS. 7, 21, and 22.” Ex-1006 ¶225. Conroy’s Figure 7, copied below, shows a power distribution table that stores power settings for various system components, and Conroy explains that these “components have *independent* throttle settings.” *Id.* ¶¶118-120.

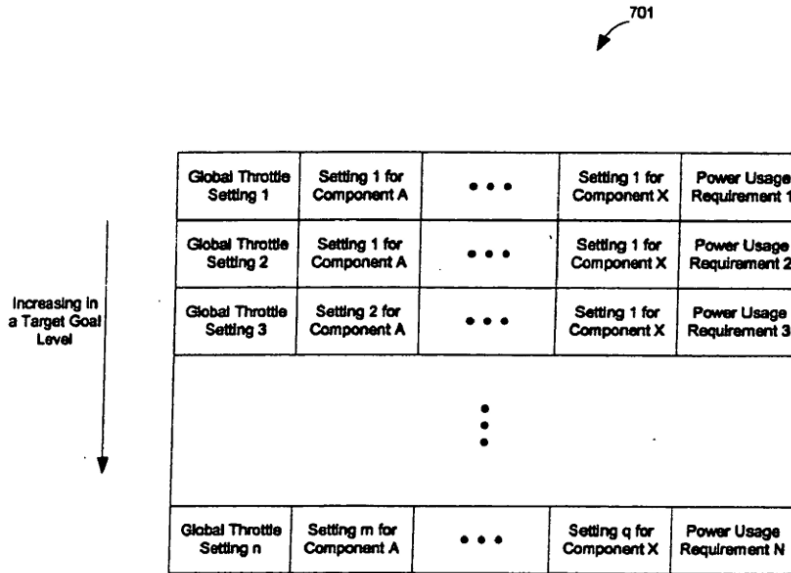


Fig. 7

Similarly, Figures 21 and 22, copied below and annotated to show different power allocations to a CPU (red) operating in a first domain and a GPU (purple) operating in a second domain, also depict power distribution tables that store independent power settings for various system components. Conroy further explains that the power units shown in these tables are only examples, and that these tables “may include various *operating frequencies or voltages* that correspond to different systems settings.” Ex-1006 ¶¶203-205; *see also id.* ¶¶50, 65-66, 81, 115-120, 188, 214, 218, 225, Figs. 7, 21, 22. Thus, a POSITA seeking to implement the teachings of Conroy would have found it obvious to operate the CPU (in a first domain) and the GPU (in a second domain) at independent voltages and frequencies so that they

can operate at independent settings for an increased ability to balance performance and control. *Id.*; *see also id.* ¶120 (“[M]ultiple components have *independent* throttle settings.”); Ex-1002 ¶¶216, 217 (explaining the relationship between operating voltages and frequencies and power settings).

Balanced load profile K1 (e.g., 0.5, 0.5)

2105	2100			
	Settings	CPU	GPU	Allocated Power
	Setting 1	A1 (e.g., 10)	B1 (e.g., 10)	P1 (e.g., 20)
	Setting 2	A2 (e.g., 7)	B2 (e.g., 7)	P2 (e.g., 14)
	Setting 3	A3 (e.g., 5)	B3 (e.g., 5)	P3 (e.g., 10)

Fig. 21 A

CPU-heavy load profile K2 (e.g., 0.75, 0.25)

2105	2110			
	Settings	CPU	GPU	Allocated Power
	Setting 1	A1 (e.g., 15)	B1 (e.g., 5)	P1 (e.g., 20)
	Setting 2	A2 (e.g., 10)	B2 (e.g., 4)	P2 (e.g., 14)
	Setting 3	A3 (e.g., 7)	B3 (e.g., 3)	P3 (e.g., 10)

Fig. 21 B

GPU-heavy load profile K3 (e.g., 0.25, 0.75)

2105	2120			
	Settings	CPU	GPU	Allocated Power
	Setting 1	A1 (e.g., 5)	B1 (e.g., 15)	P1 (e.g., 20)
	Setting 2	A2 (e.g., 3.5)	B2 (e.g., 10)	P2 (e.g., 14)
	Setting 3	A3 (e.g., 3)	B3 (e.g., 7)	P3 (e.g., 10)

Fig. 21 C

Load Profile Kn

2206	2200					
	Settings	Subsystem 1	Subsystem 2	Subsystem...	Subsystem N	Power
	1	A1	B1	...	C1	P1
	2	A2	B2	...	C2	P2
	M	Am	Bm	...	Cm	Pm

Fig. 22

**c. Element 1[B]: a memory controller coupled to the first and second domains;**

Conroy alone or in view of Finkelstein teaches Element 1[B]. Ex-1002 ¶¶219-222.

As shown in annotated Figure 37 below, Conroy discloses “system 3700” that includes “interconnect (e.g., a bus)” 3706 (green) coupled to a CPU (red) operating in a first domain and a GPU (purple) operating in a second domain. Conroy explains that “[c]omponents of the system 3700, including ... *buses*, ... are described in detail ... with respect to FIGS. 1-17.” Ex-1006 ¶225. For example, interconnect bus 1702, depicted in Figure 17, may “include one or more buses connected to each other through various bridges, *controllers* and/or adapters as is well known in the art.” *Id.* ¶187. A POSITA would have recognized that memory controllers are “controllers ... well known in the art,” and, because Conroy explicitly depicts system 3700 that includes “subsystem C (e.g., *memory*)” 3704 (orange), a POSITA seeking to implement the teachings of Conroy would have found it obvious to implement interconnect 3706 (green) by including a bus connected to a memory controller, which is in turn coupled to the CPU (red) operating in the first domain and the GPU (purple) operating in the second domain. *Id.* ¶¶187, 225, Figs. 17, 37; Ex-1002 ¶220.



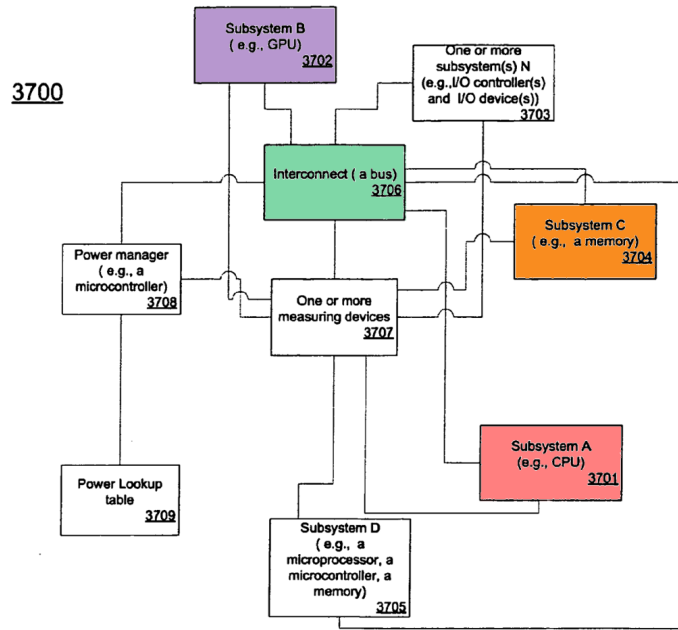


FIG. 37

Moreover, a POSITA would also have found it obvious to implement interconnect 3706 to include a bus connected to a memory controller in view of Finkelstein. Ex-1005 ¶13 (describes using “a bus ... and/or a *memory controller*” to facilitate communication with a storage device); Ex-1002 ¶221. A POSITA would have been motivated to do so to facilitate communications between various subsystems of system 3700 (including the CPU 3701 operating in the first domain and the GPU 3702 operating in the second domain) with memory 3704. Ex-1002 ¶221. Additionally, a POSITA would have had a reasonable expectation of success in doing so because such implementations are expressly taught in Finkelstein. *Id.*

**d. Element 1[C]: at least one interface; and**

Conroy teaches Element 1[C]. Ex-1002 ¶¶223-225.

As shown in annotated Figure 37, Conroy discloses “interconnect (e.g., a bus)” 3706 (green) that “interconnects ... various components together.” Ex-1006 ¶¶187, 225. A POSITA would have understood Conroy as teaching an “interface” because an “interconnection” that interconnects the various components of system 3700 together serves as an interface between these components. *Id.* ¶¶187, 225; Ex-1002 ¶224.

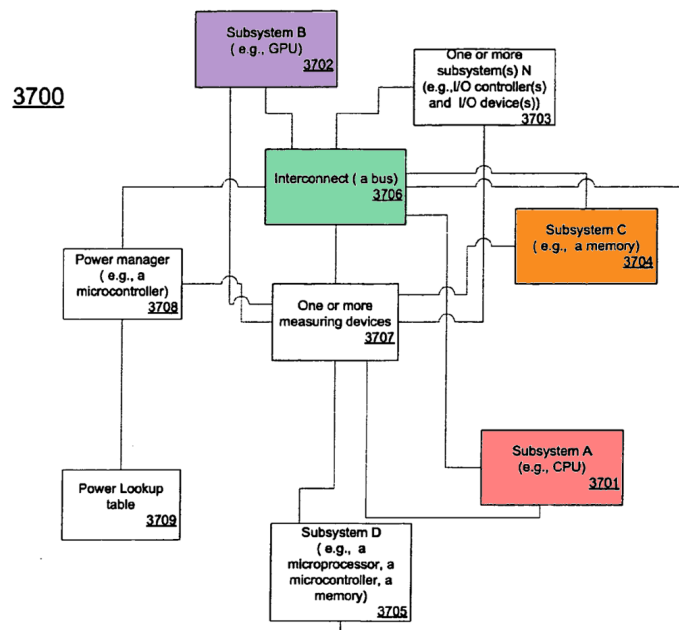


FIG. 37

- e. **Element 1[D.i]: first logic to dynamically allocate a power budget for the processor between the first and second domains at run time**

Conroy teaches Element 1[D.i]. Ex-1002 ¶¶226-229.

Conroy discloses a first logic to dynamically allocate a power budget for the processor between the first and second domains. *See* Section XII.E.1.e.

Conroy additionally discloses that this dynamic allocation of the power budget is performed “at run time.” Conroy discloses implementations in which “a workload of a subsystem determines the amount of power used by the subsystem” and explains that the power may be “redistributed in an asymmetric fashion, tracking the workloads of each of the subsystems.” Ex-1006 ¶¶196-197. For example, during a “CPU-heavy load profile,” when the workload in the first domain is intensive, power may be distributed such that “CPU consumes about 75%, while GPU consumes about 25% of the power allocated for the system,” and vice versa. *Id.* ¶203. Because Conroy discloses dynamically allocating its power budget based on the workloads of each of the subsystems, a POSITA would have understood that such an allocation is performed “at run time” when the subsystems are processing their respective workloads. Ex-1002 ¶228.

- f. Element 1[D.ii]: according to a first sharing policy value for the first domain and a second sharing policy value for the second domain, the first and second sharing policy values controllable by user-level software.**

Conroy in view of Finkelstein teaches Element 1[D.ii]. Ex-1002 ¶¶230-233.

Conroy discloses dynamically allocating a power budget according to first and second sharing policy values for the first and second domains. *See* Section XII.E.1.f.

A POSITA seeking to implement the teachings of Conroy would have found it obvious to make the first and second sharing policy values “controllable by

user-level software” in view of Finkelstein. Ex-1002 ¶232. Like Conroy, Finkelstein discloses distributing a power budget between first and second domains. *See* Sections XII.A.1.e-f. Finkelstein also discloses “user (or application defined) preferences” that describe how to distribute the power budget between the first and second domains. *Id.* A POSITA would have been motivated to combine the teachings of Conroy with the teachings of Finkelstein to provide a user with abilities to arrange or adjust the settings recorded in Conroy’s power distribution tables, which are also used to describe how to distribute a power budget between first and second domains to provide an additional level of control over the system. Ex-1002 ¶232. Additionally, a POSITA would have had a reasonable expectation of success in doing so because Conroy itself contemplates certain levels of designer/user control over the recorded settings. Ex-1006 ¶¶121, 225 (explaining that in certain implementations, the settings can be arranged “manually” by “the designer of the system”); Ex-1002 ¶232. Also, a POSITA would have had a reasonable expectation of success because Finkelstein already teaches such implementations with user-level control of power settings. Ex-1002 ¶232. Specifically, a POSITA implementing Conroy with the user-level control teachings of Finkelstein would have provided a user with the ability to input and/or change the values in Conroy’s “power distribution tables” of Figures 21 and 22. *Id.* This would have been nothing more

than providing a routine user interface to edit and/or revise values in a routine table.

*Id.*

**3. Dependent claims 2-5 and 7**

- a. Claim 2: The processor of claim 1, wherein the first logic is to determine a portion of the power budget to allocate to the first domain based on the first sharing policy value.**

Conroy teaches Claim 2. *See* Sections XII.G.2.e-f; Ex-1002 ¶¶226-234.

- b. Claim 3: The processor of claim 1, wherein the first logic is to dynamically allocate the power budget further according to a first minimum reservation value for the first domain and a second minimum reservation value for the second domain, the first and second minimum reservation values controllable by the user-level software.**

Conroy alone or in view of Finkelstein teaches Claim 3. Ex-1002 ¶¶235-238.

As discussed in Section XII.E.2.c, Conroy teaches Claim 3. Moreover, a POSITA seeking to implement the teachings of Conroy would also have found this feature obvious in view of Finkelstein, which discloses dynamically allocating a power budget according to first and second minimum reservation values for the first and second domains, and that the first and second minimum reservation values are controllable by the user-level software. *See* Section XII.A.2.b; Ex-1002 ¶¶236-237. A POSITA would have been motivated to combine the teachings of Finkelstein with the teachings of Conroy to ensure that Conroy's "minimum possible power consumption" takes into account the minimum possible power consumption of each

subsystem (including subsystems operating in the first and second domains). Ex-1006 ¶¶6, 8, 53, 198, 202-206, Abstract, Fig. 1; Ex-1002 ¶237. A POSITA would also have had a reasonable expectation of success in doing so because Conroy explicitly discloses meeting a “minimum possible power consumption” required by a system that includes subsystems such as CPU (operating in the first domain) and GPU (operating in the second domain). Ex-1006 ¶¶6, 8, 53, 198, 202-206; Ex-1002 ¶¶237.

- c. **Claim 4: The processor of claim 3, wherein the first logic is to provide at least a first portion of the power budget to the first domain, the first portion corresponding to the first minimum reservation value.**

Conroy alone or in view of Finkelstein teaches Claim 4. *See* Sections XII.G.2.e-f, XII.G.3.b; Ex-1002 ¶¶226-233, 235-239.

- d. **Claim 5: The processor of claim 1, wherein the first logic is to determine the power budget for a current time interval based at least in part on a power budget carried forward from a previous time interval, a power consumed in the previous time interval, and a power budget decay value.**

Conroy in view of Finkelstein teaches Claim 5. Ex-1002 ¶¶240-243.

Conroy discloses that the first logic determines a power budget for a current time interval based at least in part on a power consumed in the previous time interval. *See* Section XII.E.2.a.

Moreover, a POSITA seeking to implement the teachings of Conroy would have found it obvious to also take into account a power budget carried forward from a previous time interval and a power budget decay value in determining the power budget in view of Finkelstein. Ex-1002 ¶242. Like Conroy, Finkelstein discloses determining a power budget for a current time interval. *See* Section XII.A.2.d. Finkelstein also discloses an energy budget equation based at least in part on a power budget carried forward from a previous time interval, a power consumed in the previous time interval, and a power budget decay value. *Id.* A POSITA would have understood that Conroy may utilize an energy budget equation like the one disclosed in Finkelstein. Ex-1002 ¶242. For example, Conroy explains that in certain implementations, its “power budget ... is further limited according to *other conditions*.” Ex-1006 ¶74. Thus, a POSITA would have been motivated to combine the teachings of Conroy with the teachings of Finkelstein and use Finkelstein’s equation, which takes into account other conditions, to determine its power budget. Ex-1002 ¶242. Additionally, a POSITA would have had a reasonable expectation of success in using Finkelstein’s equation in Conroy’s power budget determination because both Conroy and Finkelstein are directed to managing power among multiple domains, both understand that changes to contemporary allocation may be needed based on preceding allocations, and both disclose affecting the current power

budget according to the previous cycle. Ex-1005 ¶¶28-29; Ex-1006 ¶¶12, 70-74, 80-83, 90, 120; Ex-1002 ¶242.

- e. **Claim 7: The processor of claim 1, wherein the first logic is to dynamically allocate substantially all of the power budget to the first domain for a first workload, and to dynamically allocate substantially all of the power budget to the second domain for a second workload executed after the first workload.**

Conroy teaches Claim 7.<sup>9</sup> *See* Section XII.E.2.b; Ex-1002 ¶¶190-192, 244.

#### **4. Claims 13-15 and 17-18**

Conroy teaches claims 13-15 and 17-18 as explained in Sections XII.E and XI.F. The teachings of Finkelstein would have been combined with the teachings of Conroy for same the reasons explained in Sections XII.G.1-3, and thus Conroy in view of Finkelstein teaches claims 13-15 and 17-18 for all of the reasons set forth in these prior sections. Specifically, Conroy and Finkelstein disclose interrelated teachings in the art, and a POSITA would have found any implementation details missing from Conroy in the teachings of Finkelstein, and would have been motivated to implement them to achieve the benefits taught by Finkelstein. *See also* Ex-1002 ¶¶245-255.

---

<sup>9</sup> *See supra* note 4.



### **XIII. THERE IS NO BASIS FOR ANY DISCRETIONARY DENIAL**

There is no basis for discretionary denial under §325(d) or §314(a). None of the prior art references asserted in this Petition appear in the file history of the '833 Patent. Nor has any IPR or other post-issuance challenge ever been made against the '833 Patent. Additionally, the only other proceedings relating to the '833 Patent with scheduled hearing dates are in the ITC (the concurrent District court litigations have been statutorily stayed pending the ITC proceedings). Under current USPTO guidance, concurrent ITC proceedings are not a basis to discretionarily deny IPR institution. *See* USPTO Memorandum by Katherine Vidal, June 21, 2022, *Interim Procedure For Discretionary Denials in AIA Post-Grant Proceedings with Parallel District Court Litigation*, at 5-7.

### **XIV. CONCLUSION**

The unpatentability grounds presented above are reasonably likely to prevail, *inter partes* review should be instituted, and Claims 1-5, 7, and 13-18 should be cancelled as unpatentable over the prior art.

Dated: February 2, 2023

Respectfully submitted,

/s/ Ben Haber

Ben Haber (Reg. No. 67,129)

**CERTIFICATE OF WORD COUNT**

Pursuant to 37 C.F.R. § 42.24(d), Petitioner certifies that this petition includes 13,881 words, as measured by Microsoft Word, exclusive of the table of contents, mandatory notices under § 42.8, certificates of service, word count, and exhibits.

**CERTIFICATE OF SERVICE (37 C.F.R. § 42.6(e)(1))**

The undersigned hereby certifies that the above document was served on February 2, 2023, by filing this document through the Patent Trial and Appeal Board P-TACTS System, as well as delivering a copy via express mail upon the following attorneys of record for the Patent Owner and Petitioner:

TROP, PRUNER & HU, P.C.  
PO Box 41790  
Houston, TX 77241

A courtesy copy was sent to the below counsel via electronic mail:

Andrea Leigh Fair  
Charles Everingham, IV  
Claire Abernathy Henry  
1507 Bill Owens Parkway  
Longview, TX 75604  
Phone: (903) 757-6400  
Fax: (903) 757-2323  
Email: [andrea@wsfirm.com](mailto:andrea@wsfirm.com)  
Email: [ce@wsfirm.com](mailto:ce@wsfirm.com)  
Email: [claire@wsfirm.com](mailto:claire@wsfirm.com)

Michael T. Renaud  
Mint Levin Cohn Ferris Glovasky & Popeo PC  
One Financial Center, Suite 100  
Boston, MA 02111  
Phone: (617) 542-6000  
Fax: (617) 542-2241  
Email: [mtrenaud@mintz.com](mailto:mtrenaud@mintz.com)

Brian E. Farnan  
Michael J. Farnan  
Farnan LLP  
919 North Market Street, 12th Floor

Wilmington, DE 19801  
Phone: (302) 777-0300  
Fax: (302) 777-0301  
Email: [bfarnan@farnanlaw.com](mailto:bfarnan@farnanlaw.com)  
Email: [mfarnan@farnanlaw.com](mailto:mfarnan@farnanlaw.com)

Deanna Tanner Okun  
AMS Trade LLP  
1133 Connecticut Avenue NW, Suite 1200  
Washington, DC 20036  
Phone: (202) 467-6300  
Fax: (202) 466-2006  
Email: [okun@adduci.com](mailto:okun@adduci.com)

Dated: February 2, 2023

Respectfully submitted,

/s/ Ben Haber

Ben Haber (Reg. No. 67,129)  
O'Melveny & Myers LLP  
400 South Hope Street, 18th Floor  
Los Angeles, CA 90071  
Telephone: (213) 430-6000  
Fax: (213) 430-6407  
E-Mail: [bhaber@omm.com](mailto:bhaber@omm.com)  
Attorney for Petitioner