

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

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

v.

DAEDALUS PRIME LLC
Patent Owner.

Case No. IPR2023-00661
U.S. Patent No. 10,705,960

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 10,705,960**

TABLE OF CONTENTS

	Page
I. INTRODUCTION	1
II. MANDATORY NOTICES, FEES, AND STANDING	1
A. Mandatory Notices Under 37 C.F.R. §42.8	1
B. Fee Authorization	3
C. Grounds for Standing	3
III. PRECISE RELIEF REQUESTED	3
IV. THE CHALLENGED PATENT	4
A. The '960 Patent	4
B. Level of Ordinary Skill in the Art	5
V. CLAIM CONSTRUCTION	5
VI. BRIEF DESCRIPTION OF THE APPLIED PRIOR ART REFERENCES	5
A. Moses (Ex-1005)	5
B. Balakrishnan (Ex-1006)	7
C. Trautman (Ex-1007)	9
D. Hughes (Ex-1008)	10
VII. DETAILED EXPLANATION OF THE UNPATENTABILITY GROUNDS	11
A. Ground 1: Claims 1–4, 6, 11–17, 20, 21 Are Obvious Over Moses in view of Trautman	12
1. A POSITA would have been motivated to combine Moses with Trautman	12
2. Independent Claims 1, 15, 20	14
3. Dependent Claims 2–4, 6, 11–14, 16–17, 21	31
B. Ground 2: Claims 1–3, 6, 11, 15, 16, 20, 21 Are Obvious Over Balakrishnan in View of Trautman	41

TABLE OF CONTENTS
(continued)

	Page
1. A POSITA would have been motivated to combine Balakrishnan with Trautman.....	42
2. Independent Claims 1, 15, 20	43
3. Dependent Claims 2, 3, 6, 7, 9, 11, 16, 18, 21.....	60
C. Ground 3: Claims 4, 11–14, 17 Are Obvious Over Balakrishnan in View of Trautman and Hughes	65
1. A POSITA would have been motivated to combine Balakrishnan with Trautman and Hughes.....	65
2. Claim 4/Claim 17	67
3. Claim 11	69
4. Claim 12/Claim 13/Claim 14.....	70
VIII. THE BOARD SHOULD NOT USE ITS DISCRETION TO DENY INSTITUTION UNDER <i>FINTIV</i>	73
IX. CONCLUSION	77

LIST OF EXHIBITS¹

Ex-1001	U.S. Patent No. 10,705,960 to Hum (“the ’960 Patent”)
Ex-1002	Declaration of Dr. Paul Franzon
Ex-1003	Curriculum Vitae of Dr. Paul Franzon
Ex-1004	Prosecution History of the ’960 Patent (App. No. 15/947,830)
Ex-1005	U.S. Patent Application Publication No. US2012/0173907 to Moses (“Moses”)
Ex-1006	U.S. Patent Application Publication No. US2010/0275049 to Balakrishnan (“Balakrishnan”)
Ex-1007	U.S. Patent Application Publication No. US2010/0077236 to Trautman (“Trautman”)
Ex-1008	U.S. Patent Application Publication No. US2006/0143384 to Hughes (“Hughes”)
Ex-1009	U.S. Patent Application Publication No. US2013/0268943 to Solihin (“Solihin”)
Ex-1010	U.S. Patent Application Publication No. US2009/0271646 to Talwar (“Talwar”)
Ex-1011	David Money Harris and Sarah L. Harris, <i>Digital Design and Computer Architecture</i> (2007)
Ex-1012	Rajeev Balasubramonian et al., <i>Multi-Core Cache Hierarchies</i> (2011)
Ex-1013	R. Kuppuswamy et al., <i>Over 1 million TPCC with a 45 nm 6-core Xeon CPU</i> , IEEE International Solid-State Circuits Conference (2009)
Ex-1014	U.S. Patent Application Publication No. US2011/0125971 to Kruglick (“Kruglick”)
Ex-1015	U.S. Patent Application Publication No. US2006/0004942 to Hetherington (“Hetherington”)
Ex-1016	U.S. Patent Application Publication No. US2012/0221795 to Hoshaku (“Hoshaku”)
Ex-1017	U.S. District Courts – Combined Civil and Criminal Federal Court Management Statistics (June 30, 2022), available at:

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

	https://www.uscourts.gov/statistics-reports/federal-court-management-statistics-june-2022
Ex-1018	John L. Hennessy and David A. Patterson, <i>Computer Architecture: A Quantitative Approach</i> (Elsevier 2012)
Ex-1019	Lei Jin et al., <i>A Flexible Data to L2 Cache Mapping Approach for Future Multicore Processors</i> , University of Pittsburgh (2006)
Ex-1020	Claim Comparison Table
Ex-1021	Nikos Hardavellas et al., <i>Reactive NUCA: Near-Optimal Block Placement and Replication in Distributed Cache</i> , ISCA (2009)
Ex-1022	U.S. Patent Application Publication No. US2010/0274973 to Balakrishnan (“Balakrishnan-2”)
Ex-1023	U.S. Patent No. 8,892,916 to Bieswanger (“Bieswanger”)
Ex-1024	U.S. Patent Application Publication No. US2010/0005323 to Kuroda (KR 2008/114835) (“Kuroda”)
Ex-1025	U.S. Patent Application Publication No. US2011/0276812 to Lee (KR 2011/123698) (“Lee”)
Ex-1026	Michael Keating et al., <i>Low Power Methodology Manual: For System-on-Chip Design</i> (Springer 2008)
Ex-1027	Stefan Rusu et al., <i>Power Reduction Techniques for an 8-core Xeon® Processor</i> , IEEE (2009)
Ex-1028	U.S. Patent Application Publication No. US2008/0016319 to Pappalardo (“Pappalardo”)
Ex-1029	U.S. Patent Application Publication No. US2013/0246825 to Shannon (“Shannon”)
Ex-1030	PCI Express Base Specification Revision 1.0 (2002)
Ex-1031	P. Franzon, <i>Fault Tolerance in VLSI</i> , University of Adelaide (1988)
Ex-1032	Riccardo Paccagnella et al., <i>Lord of the Ring(s): Side Channel Attacks on the CPU On-Chip Ring Interconnect Are Practical</i> , USENIX Security Symposium (2021)

TABLE OF CONTENTS

Page

I. INTRODUCTION

Samsung Electronics Co., Ltd., and Samsung Electronics America, Inc., (“Petitioner”) request *inter partes* review (“IPR”) of Claims 1–4, 6, 11–17, 20, 21 of U.S. Patent No. 10,705,960 (“the ’960 Patent”) (Ex-1001), currently assigned to Daedalus Prime LLC (“PO”).

II. MANDATORY NOTICES, FEES, AND STANDING

A. 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., Samsung Semiconductor, Inc., Samsung Austin Semiconductor, LLC, and Arm Ltd.

Related Matters: PO has asserted the ’960 Patent against the real parties-in-interest in *Daedalus Prime LLC v. Samsung Elecs. Co., Ltd.*, No. 2:22-cv-00354 (E.D. Tex. Sept. 12, 2022).

Lead and Back-Up Counsel:

- Lead Counsel:

John Kappos (Reg. No. 37,861)
O’Melveny & Myers LLP
2501 North Harwood Street, Suite 1700
Dallas, TX 75201
Telephone: (972) 360-1900
Fax: (972) 360-1901
E-Mail: jkappos@omm.com

- First Backup Counsel:

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

- Additional Backup Counsel:

Xin-Yi Zhou (Reg. No. 63,366)
O'Melveny & Myers LLP
400 South Hope Street, 18th Floor
Los Angeles, CA 90071
Telephone: (213) 430-6000
Fax: (213) 430-6407
E-Mail: vzhou@omm.com

Jeff Baxter (Reg. No. 45,560)
O'Melveny & Myers LLP
2501 North Harwood Street, Suite 1700
Dallas, TX 75201
Telephone: (972) 360-1900
Fax: (972) 360-1901
E-Mail: jbaxter@omm.com

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

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

- jkappos@omm.com
- bhaber@omm.com
- vzhou@omm.com
- jbaxter@omm.com
- tfink@omm.com
- OMMSAMSUNGDEADALUSEDTX@omm.com

B. Fee Authorization

The PTO is authorized to charge any fees due during this proceeding to Deposit Account No. 50-0639.

C. Grounds for Standing

Petitioner certifies that the '960 Patent is available for review, and Petitioner is not barred or estopped from requesting review.

III. PRECISE RELIEF REQUESTED

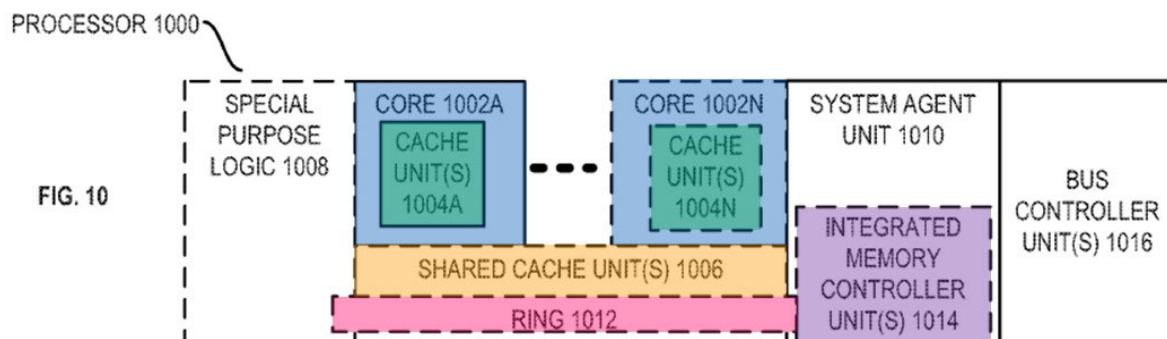
Petitioner requests cancellation of the challenged claims based on the following grounds:

Ground	Summary
1	Claims 1–4, 6, 11–17, 20, 21 are obvious over Moses (Ex-1005) in view of Trautman (Ex-1007).
2	Claims 1–3, 6, 11, 15, 16, 20, 21 are obvious over Balakrishnan (Ex-1006) in view of Trautman.
3	Claims 4, 11–14, 17 are obvious over Balakrishnan in view of Trautman and Hughes (Ex-1008).

IV. THE CHALLENGED PATENT

A. The '960 Patent

The '960 Patent discloses processors having clustered cores and cache slices, methods performed by and/or within the processors, and systems incorporating one or more processors. Ex-1001, 2:43–46. Fig. 10 (annotated) illustrates a processor 1000 including a plurality of symmetric, multi-threaded cores 1002A–N; a cache subsystem including a lower-level cache 1004A–N and a shared cache 1006 that is physically distributed across a die; cache management circuitry, such as ring-based interconnect 1012 capable of providing coherent, non-uniform access between one or more cache units 1006 and cores 1002-A–N; power management circuitry capable of controlling frequency and gating power at the cluster level, in which an average distance between cores in the first cluster is less than an average distance between the plurality of cores; and a first and second integrated memory controller 1014. *Id.* at 17:46–18:42.



B. 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 engineering, or a related field, and at least three years of experience in the research, design, or development of processor systems, or the equivalent, with additional education substituting for experience and vice versa. Ex-1002 ¶¶33.

V. CLAIM CONSTRUCTION

Petitioner interprets the ’960 Patent’s claims according to the *Phillips* claim construction standard. 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.² Ex-1002 ¶¶34–35.

VI. BRIEF DESCRIPTION OF THE APPLIED PRIOR ART REFERENCES

This petition presents four references, none of which were of record during prosecution. *See* Ex-1004.

A. Moses (Ex-1005)

Moses was filed 12/30/2011, and published 7/5/2012. Ex-1005. Moses qualifies as prior art under at least pre-AIA 35 U.S.C. § 102(a). Ex-1002 ¶¶63–65.

² Petitioner reserves all rights to raise claim construction and other arguments in district court. Additionally, Petitioner will request leave to submit the district court’s claim construction when available.

Moses discloses “a system-on-chip (SoC) processor,” including “one or more central processing unit (CPU) cores 510.” Ex-1005, Fig. 5, ¶55. “Each CPU core 510 also has its own core cache (not shown), for example a level 1 (L1) cache, which is not shared by the other CPU cores” and cache 540, which “is a shared cache that is shared among the CPU cores 510, and can be accessed by any of the CPU cores 510.” *Id.*

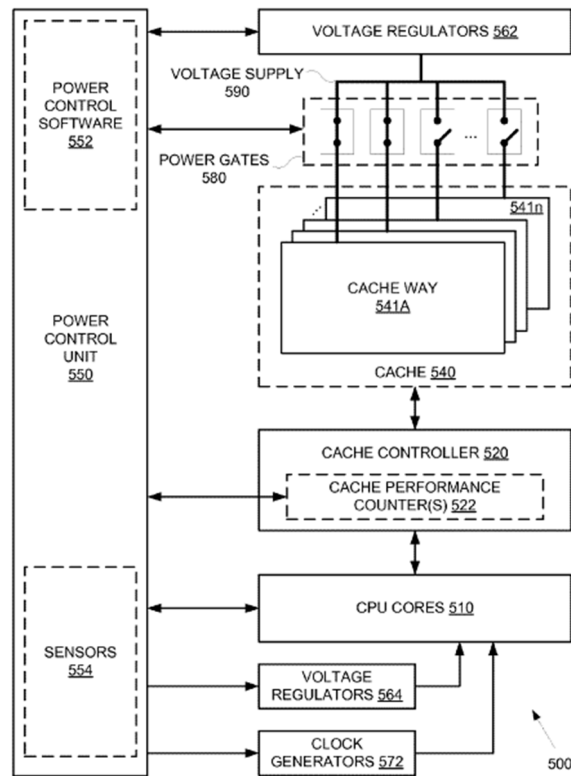
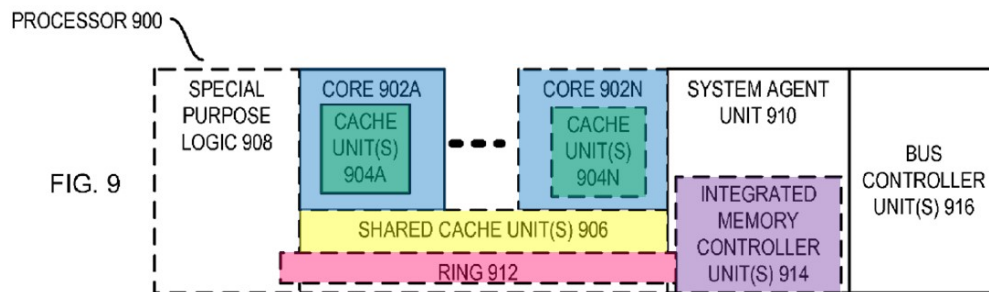


FIG. 5

Id., Fig. 5.

Figure 9 illustrates a “processor 900” having “multiple cores 902A–N” (blue) and “a set of one or more integrated memory controller unit(s) 914” (purple).” Ex-1005 ¶92. “The memory hierarchy includes one or more levels of

cache within the cores,” including cache unit(s) 904A–N (green), “a set or one or more shared cache units 906 [(yellow)], and external memory (not shown) coupled to the set of integrated memory controller units 914.” *Id.* ¶94. “[A] ring based interconnect unit 912 [(pink)] interconnects the integrated graphics logic 908, the set of shared cache units 906, and the system agent unit 910/integrated memory controller unit(s) 914.” *Id.*



Id., Fig. 9.

B. Balakrishnan (Ex-1006)

Balakrishnan was filed 4/24/2009, and published 10/28/2010. Ex-1006.

Balakrishnan qualifies as prior art under at least pre-AIA 35 U.S.C. § 102(b). Ex-1002 ¶¶66–68.

Balakrishnan discloses “non-uniform cache access (NUCA) systems.” Ex-1006 ¶1. Balakrishnan discloses “banks of a non-uniform cache access (NUCA) cache, with the ways being vertically distributed across multiple banks.” *Id.* ¶10. Further, Balakrishnan discloses “a computing device, having one or more processors coupled with one or more NUCA cache elements.” *Id.* ¶26. “The

embodiments comprise switches configured to turn off groups of banks.... State selectors may be coupled to the switches to select different power states of the NUCA caches.” *Id.* ¶10.

Balakrishnan’s Figure 1 illustrates “system 100 [having] four processors, 105, 110, 115, and 120.” Ex-1006 ¶29. “For example, processor 105 comprises two cores, 125 and 130 [(blue)],” which “may also each comprise L1 cache [(green)].” *Id.* ¶30. “System 100 may also have one or more L2 cache elements, such as NUCA cache 135. NUCA cache 135 may comprise a number of banks, such as bank 140.” *Id.* “The processors of system 100 may be connected to other components via a system or fabric bus 180,” including “coupl[ing] processors 105, 110, 115, and 120 to system memory 175” and shared “L3 cache 170” (yellow). *Id.* ¶34.

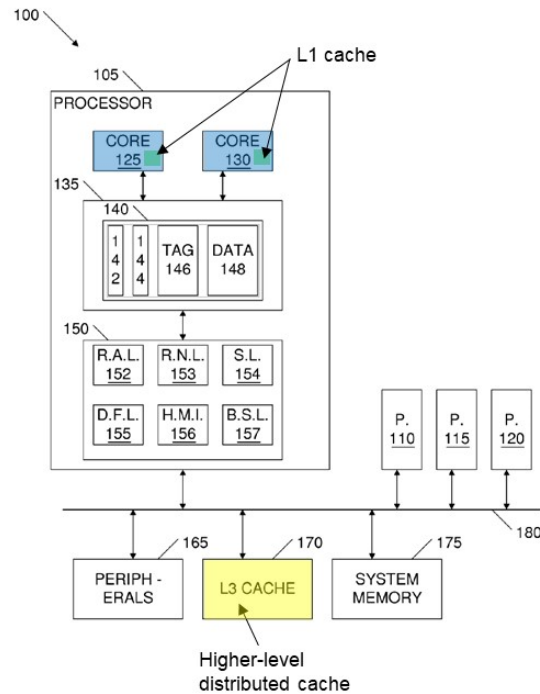


FIG. 1

Id., Fig. 1.

C. Trautman (Ex-1007)

Trautman was filed 11/23/2009, and published 3/25/2010. Trautman qualifies as prior art under at least pre-AIA 35 U.S.C. § 102(b). Ex-1002 ¶¶69–70.

Trautman discloses “dynamically distributing” a computational load to different cores of a multi-core processor. Ex-1007. Trautman discloses clustering cores and operating a core cluster “at a higher frequency than the cores not included in the cluster since the cores of [the] cluster...have the computational load being processed by the cluster dynamically distributed amongst the cores of the cluster.” *Id.* ¶¶44–47. In addition, Trautman discloses that “a DTM [(dynamic thermal management)] mechanism...may be invoked to dynamically distribute the

processing of a computational load across the clusters” or “may be used to distribute the processing between clusters.” *Id.* ¶49.

D. Hughes (Ex-1008)

Hughes was filed 12/27/2004, and published 6/29/2006. Hughes thus qualifies as prior art under pre-AIA 35 U.S.C. § 102(a), (b), and (e). Ex-1002 ¶¶71–73.

Hughes discloses “a distributed shared cache in a multi-core processor.” Ex-1008. Specifically, Hughes teaches that “[p]rocessor 100 may include several processor cores 102–116 [(blue)] and cache molecules 120–134 [(yellow)],” where “[t]he cache molecules 120–134 collectively may be functionally equivalent to a traditional unitary cache. In one embodiment, they may form a level two (L2) cache, with a level one (L1) cache [(green)] being located within cores 102–116.” *Id.* ¶17.

In addition, “[t]he cores 102–116 and cache molecules 120–134 are shown connected with a redundant bi-directional ring interconnect, consisting of clockwise (CW) ring 140 [(purple)] and counter-clockwise (CCW) ring 142 [(purple)].” Ex-1008 ¶18. Although “[e]ach portion of the ring may convey any data among the modules shown,” Hughes teaches that “[e]ach core of core 102–116 is shown being paired with [one] of cache molecules 120–134.” *Id.* “The pairing is to logically associate a core with the ‘closest’ cache molecule in terms of

low access latency,” and may specifically consider “[a] metric of ‘distance’ ...to describe a latency ordering of cache molecules with respect to a particular core.”

Id.

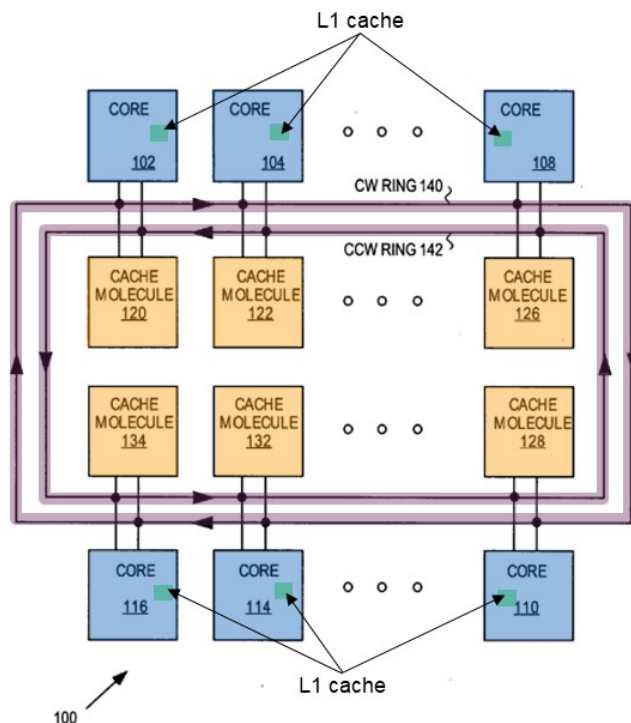


FIG. 1

Id., Fig. 1.

VII. DETAILED EXPLANATION OF THE UNPATENTABILITY GROUNDS

The '960 Patent contains 21 claims. Independent Claims 1, 15, and 20 are substantially similar. Like claim elements are discussed together in the element-by-element analysis. *See* Ex-1020 (Claim Comparison Table). The subject matter of the challenged claims is disclosed by the prior art. Ex-1002 ¶74.

A. Ground 1: Claims 1–4, 6, 11–17, 20, 21 Are Obvious Over Moses in view of Trautman

The subject matter of the challenged claims is disclosed by Moses alone or in view of other secondary references. Moses’s Figures 5, 8A, and 9 depict most of the limitations of the challenged claims and should be understood as “like references indicat[ing] similar elements.” *See* Ex-1005 ¶¶4, 9, 13, 15. Thus, a POSITA would understand that the systems depicted in Moses’s Figures 5, 8A, and 9 represent the same embodiment or compatible systems with similar elements. Ex-1002 ¶75.

1. A POSITA would have been motivated to combine Moses with Trautman

A POSITA would have been motivated to combine Moses and Trautman and would have had a reasonable expectation of success in doing so, because they are both from the same field and relate to the same well-known technologies. Ex-1002 ¶76. Moses and Trautman provide complementary solutions for power management of multi-core processors. *Compare* Ex-1005 ¶¶3, 23–26, *with* Ex-1007 ¶¶15–18. A POSITA seeking to increase system efficiency in a multi-core processor would have known of and consulted these proposed solutions. Ex-1002 ¶76.

Moses discloses techniques for power conservation in multi-core systems with cache subsystems. Ex-1005 ¶¶2–3, 23–26. Trautman provides additional well-

known teachings of dynamic thermal management to reduce leakage power of a processor. Ex-1007 ¶18. A POSITA would have been motivated to modify Moses to decrease power consumption by considering temperature and power density and improved power management through clustering. These benefits improve efficiency, which lowers packaging and operating costs, extends battery life, and extends useful life of the products. Ex-1002 ¶77; *see* Ex-1007 ¶¶15, 18, 58; *see also* Ex-1006 ¶¶7, 9–12; *Dystar Textilfarben GmbH v. C.H. Patrick Co.*, 464 F.3d 1356, 1368 (Fed. Cir. 2006) (“[I]mplicit motivation to combine exists...when the ‘improvement’ is technology-independent and the combination of references results in a product or process that is more desirable...because it is stronger, cheaper, cleaner, faster, lighter, smaller, more durable, or more efficient.”).

A POSITA would have had a reasonable expectation of success in combining the teachings of Moses and Trautman with no change in their respective functions because each purports to solve these problems in similar and predictable ways by using well-known implementations and techniques, such as thread migration and clustering. Ex-1002 ¶¶78.

2. Independent Claims 1, 15, 20³

a. Element 1[preamble]: A system comprising:

Element 15[preamble]: A method comprising:

Element 20[preamble]: A machine comprising:

Moses describes “[e]mbodiments of systems, apparatuses, and methods for energy-efficient operation of a device.” Ex-1005, code (57); *see also id.* ¶¶23, 31, 53, 55, 67–69, 116–19, Figs. 2, 4; Ex-1002 ¶79.

b. Element 20[a]: a processor comprising:

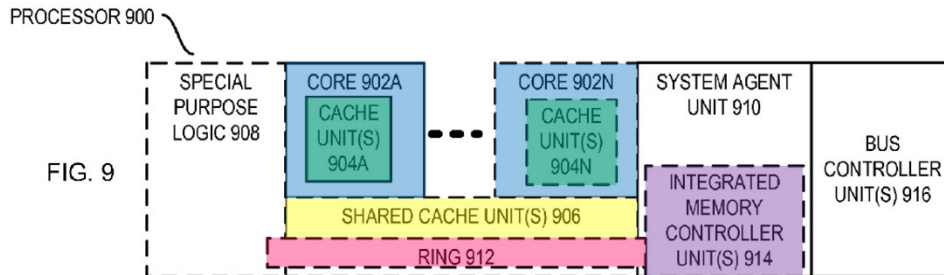
Moses’s Figure 5 “illustrates a block diagram of a system 500.... [S]ystem 500 is a system-on-chip (SoC) processor. The system 500 includes one or more central processing unit (CPU) cores 510.” Ex-1005 ¶¶55, 65, 92, Fig. 9; Ex-1002 ¶108.

c. Elements 1[a]/15[a]/20[b]: [executing instructions and processing data with] a plurality of cores, the plurality of cores comprising symmetric multi-threaded cores;

First, Moses discloses a plurality of cores. Moses’s Figure 5 illustrates system 500, which “is a system-on-chip (SoC) processor,” including “one or more central processing unit (CPU) cores 510.” Ex-1005 ¶55. In addition, Moses’s

³ Although the claim language of independent Claims 1, 15, and 20 differ slightly, the claim requirements are substantively identical and any meaningful differences are analyzed.

Figure 9 (annotated) depicts “a processor that may have more than one core [902A–N (blue)].” *Id.* ¶92.



Id., Fig. 9.

Second, Moses discloses “executing instructions and processing data” (Claim 15) with a plurality of cores:

The DSU [data storage unit] 620 may include a machine-accessible medium 607 on which is stored one or more sets of instructions (e.g., software) embodying any one or more of the methods or functions described herein.... The software may also reside, completely or at least partially, within the processor(s) 605 during execution thereof by the processor(s) 605.

Ex-1005 ¶67. Thus, Moses discloses executing instructions and processing data with a plurality of cores.

Third, Moses discloses that “cores 902A–N are capable of multi-threading.” Ex-1005 ¶95; *see id.* ¶84 (“[T]he core may support multithreading (executing two or more parallel sets of operations or threads) and may do so in a variety of ways including...simultaneous multithreading (where a single physical core provides a logical core for each of the threads that core is simultaneously multithreading).”).

Finally, Moses discloses symmetric cores:

cores 902A–N may be homogenous...in terms of architecture instruction set; that is, two or more of the cores 902A–N may be capable of execut[ing] the same instruction set....

Id. ¶96; *see also id.* ¶¶89, 94. A POSITA would have understood Moses’s homogeneous architecture to teach symmetric cores. Ex-1002 ¶¶80–81, 99, 109; *see* Ex-1001, 18:37–41 (“The cores 102 may be symmetric cores, asymmetric cores (e.g., heterogeneous cores), or some may be symmetric and others asymmetric.”). Thus, Moses discloses a plurality of symmetric, multi-threaded cores.

d. Elements 1[b]/15[b]/20[c]: [storing the instructions and the data in] a cache subsystem,

Moses’s Figure 7B illustrates a processor with “separate instruction and data cache units 734/774 and a shared L2 cache unit 776, alternative embodiments may have a single internal cache for both instructions and data, such as, for example, a level 1 (L1) internal cache, or multiple levels of internal cache.” Ex-1005 ¶85; *see also id.* ¶67 (“The software [(i.e., sets of instructions)] may also reside, completely or at least partially, within the processor(s) 605 during execution thereof by the processor(s) 605.”). Moses also discloses that “[t]he memory hierarchy includes one or more levels of cache within the cores, a set or one or more shared cache units 906, and external memory...coupled to the set of integrated memory

controller units 914.” *Id.* ¶¶94; Ex-1002 ¶¶82, 100, 110.

- e. **Elements 1[c]/15[c]/20[d]: the cache subsystem comprising a plurality of first level caches and at least one higher level distributed cache comprising a plurality of distributed cache portions that are physically distributed across a die and shared by the plurality of cores, each first level cache integral to one of the plurality of cores and each distributed cache portion accessible to each of the plurality of cores;**

Moses discloses that “[t]he memory hierarchy includes one or more levels of cache within the cores, a set or one or more shared cache units 906, and external memory...coupled to the set of integrated memory controller units 914.” Ex-1005 ¶94. For example, “[t]he set of shared cache units 906 may include one or more mid-level caches, such as level 2 (L2), level 3 (L3), level 4 (L4), or other levels of cache, a last level cache (LLC), and/or combinations thereof.” *Id.* Moses discloses a system, in which “[e]ach CPU core 510 has its own core cache..., for example, a level 1 (L1) cache, which is not shared by the other CPU cores 510.” *Id.* ¶88. Because the L1 cache is private to each core, a POSITA would have understood or found it obvious that Moses discloses at least one first level cache integral to one of the plurality of cores. Ex-1002 ¶¶83–86, 101, 111.

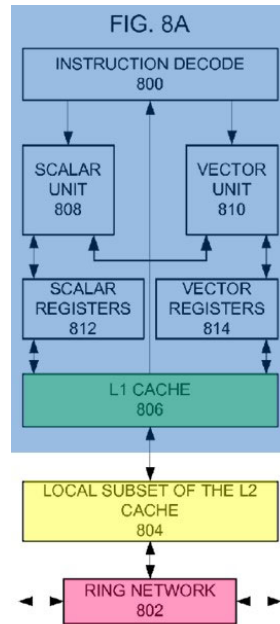
In addition, “[t]he system also includes a cache 540 that is coupled to the CPU cores 510.” Ex-1005 ¶55. “The cache 540...can be a last level cache (LLC), or a shared mid-level cache such as a shared level 2 (L2), level 3 (L3), level 4 (L4), or other levels of shared data cache and/or combinations thereof. The cache 540 is

organized into a plurality of cache partitions.” *Id.* ¶¶56, 94. Moreover, “[t]he cache 540 is a shared cache,” in which the plurality of cache partitions are “shared among the CPU cores 510, and can be accessed by any of the CPU cores 510.” *Id.* ¶55.

As discussed above, Moses’s Figure 8A (annotated) depicts “a single processor core [(blue)], along with its connection to the on-die interconnect network 802 [(pink)],” “[a]n L1 cache 806 [(green)] allow[ing] low-latency accesses to cache memory,” and “its local subset of the Level 2 (L2) cache [(yellow)].” Ex-1005 ¶88. Moses explains:

The local subset [or portion] of the L2 cache 804 is part of a global L2 cache [or shared cache, such as cache 540] that is divided into separate local subsets [or portions], one per processor core. Each processor core has a direct access path to its own local subset of the L2 cache 804.

Id. ¶89.



Id., Fig. 8A; *see also id.*, Fig. 9 (similarly showing cache portions 904A–N integral to cores 902A–N and shared cache 906 shared between cores 902A–N). Therefore, Moses discloses at least one higher level distributed cache comprising a plurality of distributed cache portions.

Moreover, a POSITA would have understood—or at least found it obvious—that Moses’s distributed cache portions are distributed across the die and accessible to each of the plurality of cores. Ex-1002 ¶¶83–86, 101, 111. The “global L2 cache...is divided into separate local subsets [or portions], one per processor core,” so a POSITA would have understood that the local portions of the L2 cache are distributed across the die according to the location of each processor. Ex-1005 ¶89; Ex-1002 ¶¶83–86, 101, 111. Although each processor core has its own cache portion and direct access to it, a POSITA would have understood—or at

least found it obvious—that Moses’s ring network provides each of the cores access to each of the cache portions because “[t]he ring network ensures coherency for shared data. Ex-1002 ¶¶83–86, 101, 111. The ring network is bidirectional to allow agents such as processor cores, L2 caches and other logic blocks to communicate with each other within the chip.” Ex-1005 ¶89; *see also, e.g.*, Ex-1008 ¶¶3, 18. Thus, Moses either discloses that each distributed cache portion is shared by—and therefore accessible to—each of the plurality of cores via the ring network or renders it obvious.

f. Elements 1[d]/15[d]/20[e]: cache management circuitry operative to provide coherent, non-uniform access to the plurality of distributed cache portions by the plurality of cores;

Moses describes that “[e]ach processor core has a direct access path to its own local subset of the L2 cache 804.” Ex-1005 ¶89. Moses discloses that “[d]ata read by a processor core is stored in its L2 cache subset 804 and can be accessed quickly, in parallel with other processor cores accessing their own local L2 cache subsets.” *Id.* ¶89. Moses also states that a “ring network ensures coherency for shared data. The ring network is bidirectional to allow agents such as processor cores, L2 caches and other logic blocks to communicate with each other within the chip.” *Id.* Because Moses discloses quick, direct access (i.e., lower latency) by a core to a local cache portion and bidirectional access over a ring network to the other L2 cache portions, a POSITA would have understood that Moses discloses

non-uniform access to the plurality of distributed cache portions by the plurality of cores. Ex-1002 ¶¶87–89, 102, 112; *see, e.g.*, Ex-1008 ¶¶3, 18.

A POSITA would have understood that Moses discloses or renders obvious non-uniform cache access in light of a POSITA’s knowledge. Ex-1002 ¶¶87–89, 102, 112. For example, a POSITA would have known that “[i]n NUCA caches, the latency to a bank generally depends on the proximity to the device making the request, which frequently is a processor.” *See, e.g.*, Ex-1006 ¶7; Ex-1002 ¶¶87–89, 102, 112. A POSITA also would have known that “NUCA allows banks closest to the processor to respond the fastest and forces the banks furthest from the processor to respond the slowest.” Ex-1006 ¶7; Ex-1002 ¶¶87–89, 102, 112. Thus, a POSITA would have understood that “[t]he distance hierarchy may refer to the several levels of delay or access time,” including “accumulated delays caused by interconnections, connecting wires,...etc.” Ex-1006 ¶31; Ex-1002 ¶¶87–89, 102, 112. Because Moses discloses different access delays for a core accessing a local cache portion directly and a non-proximate cache portion over the ring network, a POSITA would have understood that Moses describes a non-uniform cache access and discloses this limitation. Ex-1002 ¶¶87–89, 102, 112.

To the extent that Moses does not disclose non-uniform access to the plurality of distributed cache portions by the plurality of cores, this limitation would have been obvious over Moses in view of the knowledge of a POSITA for

the same reasons discussed above. Ex-1002 ¶¶87–89, 102, 112. A POSITA would have been motivated to improve Moses by using non-uniform cache access to reduce latency between a core and at least one distributed cache portion, improve system performance with locally-shared memory, and enable system expansion as needs of the system evolve. *Id.* Moreover, a POSITA would have a reasonable expectation of success in doing so because non-uniform cache access was well-known in the art and would be compatible with a system comprising a shared LLC. *Id.*

- g. Element 20[f]: an article of manufacture comprising a machine-readable storage medium storing a plurality of instructions, the plurality of instructions, when performed by the machine, causing the machine to perform operations comprising:**

Moses discloses a system including “a storage unit 1128 such as a disk drive or other mass storage device which may include instructions/code” that, when executed, cause the machine to perform the claimed operations. Ex-1005 ¶110; *see also id.* ¶¶104, 115–18, Fig. 11; Ex-1002 ¶113.

- h. Elements 1[e]/15[e]/20[g]: power management circuitry operative to enable a first frequency of operation for a first cluster of the plurality of cores which are physically proximate to one another and a second frequency of operation for a second cluster of the plurality of cores which are physically proximate to one another,**

Moses discloses that “[t]he clocks of CPU cores can be driven by the output

of clock generators 572,” which “allow the clock frequency to each CPU core 510 to be adjusted depending on processor load.” Ex-1005 ¶58. “For example, to reduce power consumption, the clock frequency of a CPU core 510 can be lowered when the activity and load of that CPU core 510 is low, or the clock to the CPU core 510 can be turned off when the CPU core 510 is idle.” *Id.*; *see also id.* ¶¶54, 59, 60. Further, a POSITA would have found it obvious to modify Moses to control the frequency for a cluster of cores because the system could apply a single clock paradigm to all the cores of a cluster, making the system easier to implement and control. Ex-1002 ¶¶90–91, 103, 114.

To the extent that Moses does not explicitly disclose controlling a frequency of operation for a cluster of cores that are physically proximate to one another, it would have been obvious in view of Trautman. *Id.* Trautman’s Figure 6 (annotated) illustrates assigning cores to groups (i.e., clusters)—based, at least in part, on a physical distance or adjacency between the cores—and configuring each cluster to operate at a frequency different from cores outside each cluster.

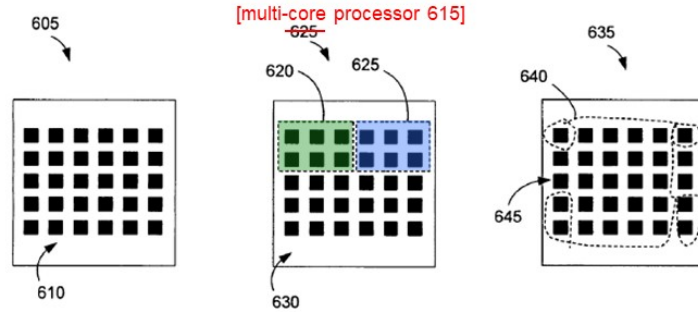


FIG. 6

Id., Fig. 6. “Clusters 620 [(green)] and 625 [(blue)] may operate at a higher frequency than the remaining cores of multi-core processor 615 [(mis]labeled as ‘625’ in the figure)] not included in clusters 620 and 625.” Ex-1007 ¶44; *id.* ¶47 (“The clustered cores may be adjoining.”); *see also id.* ¶45 (“Cores of cluster 640 may be operated at a higher frequency than the cores not included in the cluster since the cores of cluster 640 have the computational load being processed by the cluster dynamically distributed amongst the cores of the cluster.”). Thus, a POSITA would have been motivated to increase system efficiency and ease of operation within a system by modifying Moses in view of Trautman to cluster cores based on physical proximity and thereby control the frequency of the entire cluster. Ex-1002 ¶¶90–91, 103, 114; *Dystar*, 464 F.3d at 1368.

- i. **Elements 1[f]/15[f]/20[h]: wherein an average distance between cores in the first cluster is less than an average distance between the plurality of cores,**

Trautman discloses that “a cluster may include at least two cores,” which

“may be adjoining.” Ex-1007 ¶47; *see id.* ¶45. For example, Trautman’s Figure 6 depicts “a multi-core processor 615 including an array of cores 630 wherein a number of cores 630 are grouped into two clusters 620 and 625. Clusters 620 and 625 may, under the control of a DTM mechanism in accordance herewith, operate as superscalar clusters that are selected from adjacent cores 630.” *Id.* ¶44.

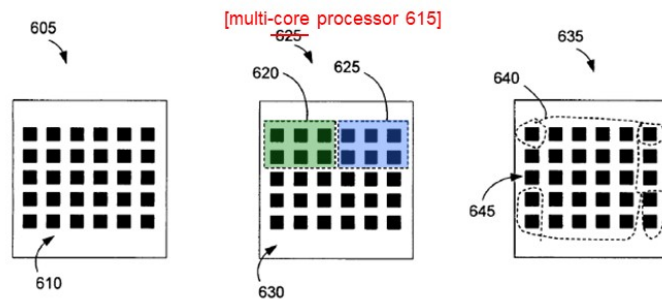


FIG. 6

Id., Fig. 6. As shown in Figure 6, clusters 620 and 625 comprise physically proximate cores. *See id.* Accordingly, a POSITA would have understood that the average distance between cores in either cluster 620 or 625 was less than the average distance between all of the cores of the array 630. Ex-1002 ¶¶92–93, 104, 115.

Further, a POSITA would have understood that clustering adjoining or physically proximate cores to achieve a lower average distance between cores within a cluster would decrease access latency within the cluster and allow for frequency control or power gating at the cluster level because this was well-known

in the art. Ex-1002 ¶¶92–93, 104, 115; *see, e.g.*, Ex-1009 ¶22. Therefore, a POSITA would have been motivated to modify Moses in view of Trautman’s teachings of clustering physically proximate cores to further decrease the access latency between cores in a cluster and increase the efficiency of the system. Ex-1002 ¶¶92–93, 104, 115; *Dystar*, 464 F.3d at 1368.

- j. **Elements 1[g]/15[g]/20[i]: [the power management circuitry operative to] selectively gate power to the first cluster of the plurality of cores and distributed cache portions [of the at least one higher level distributed cache] that correspond to the first cluster and/or the second cluster of the plurality of cores and distributed cache portions [of the at least one higher level distributed cache] that correspond to the second cluster;**

Moses discloses that “the power consumption of a multi-core processor can be reduced by monitoring the processor activity of each central processor unit (CPU) core and to power down a CPU core when that particular CPU core is idle.” Ex-1005 ¶3. Further, Moses discloses, “As each CPU core can be independently powered down into a low power state, parts of the LLC [(last level cache)] can also be powered down to shrink the LLC size to further reduce power consumption.” *Id.* “[W]ith less CPU cores that are active, less LLC is expected to be needed. Hence, in these power management techniques, the LLC is resized based on the processor activity of the CPU cores in the processor.” *Id.* Accordingly, a POSITA would have understood that a cluster of cores (e.g., all cores in a processor) could be

power gated with portions of a shared cache that correspond to the cores in the cluster. Ex-1002 ¶¶94–96, 105, 116; *see id.* ¶55 (“The system also includes a [shared] cache 540 that is coupled to the CPU cores 510. The cache 540 is a shared cache that is shared among the CPU cores 510, and can be accessed by any of the CPU cores 510.”); *see also id.* ¶53 (describing an “enhanced deep-sleep state...where all CPU cores in the processor are powered down, and the entire cache is flushed and powered down.”).

Moses’s Figure 5 (annotated) shows that “the system 500 includes a power control unit (PCU) 550 to manage the power consumption of system 500 and to control the power related components of the system 500 such as the voltage regulators 562 and 564, clock generators 572, and power gates 580 [(red)].” Ex-1005 ¶59. Figure 5 illustrates that “each cache way 541A–541 n [(yellow)] is coupled to a voltage supply 590 through power gates 580,” which allow “each cache way 541A–541 n to be independently enabled/powered-on or be disabled/powered-down to resize the cache 540.” *Id.* ¶56.

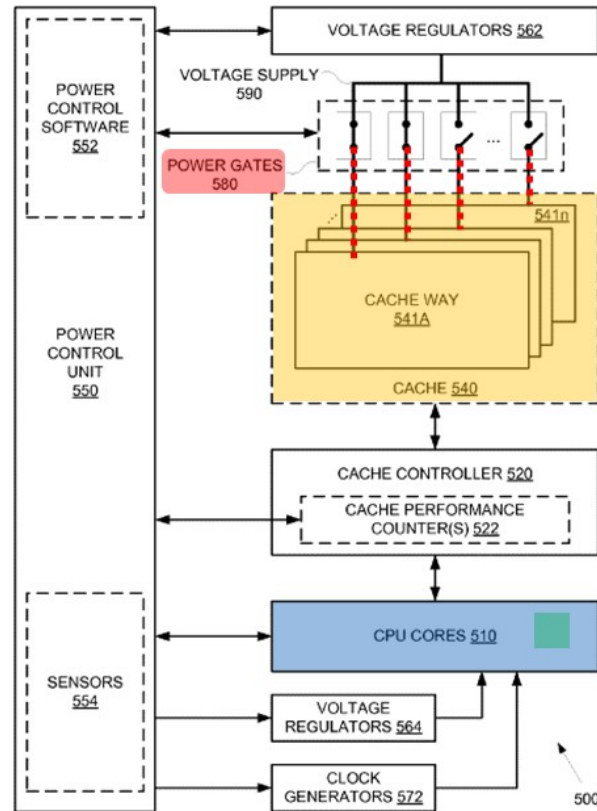


FIG. 5

Id., Fig. 5.

To the extent that Moses does not explicitly disclose this limitation, a POSITA motivated to increase efficiency of a system and ease of implementation would have found it obvious over Moses, alone or in view of Trautman's teachings of clustering and frequency control of a cluster of physically proximate cores, to selectively gate power to clusters of cores and corresponding cache portions. Ex-1002 ¶¶94–96, 105, 116; Ex-1007 ¶¶47, 44; Ex-1006 ¶¶1, 50; *Dystar*, 464 F.3d at 1368. Moses discloses power gating multiple cores of a processor together and powering down portions of the LLC based on processor activity to decrease power

consumption. Ex-1005 ¶¶3, 53. For example, a POSITA would have understood that power could be selectively gated to a cluster of CPU cores 510 and cache ways 541A–541n, or a subset thereof, that correspond to the cluster. Ex-1002 ¶¶94–96, 105, 116. Further, Trautman discloses controlling the frequency of a cluster of cores based on computational load to decrease power leakage. Ex-1007 ¶¶47, 44. Thus, a POSITA would have been motivated to combine these complementary teachings to power gate and control the frequency of a cluster of cores and its corresponding cache portions to decrease power consumption and leakage. Ex-1002 ¶¶94–96, 105, 116.

- k. **Elements 1[h]/20[j]: a first integrated memory controller coupled with the symmetric multi-threaded cores; and a second integrated memory controller coupled with the symmetric multi-threaded cores.**

Element 15[h]: controlling access by the symmetric multi-threaded cores to a first system memory with a first integrated memory controller; and controlling access by the symmetric multi-threaded cores to a second system memory with a second integrated memory controller.

Moses’s Figure 11 (annotated) depicts processors 1170 and 1180, which “may be some version of the processor 900.” Ex-1005 ¶105. “[P]rocessor 900...may have more than one core, may have an integrated memory controller,” and a “memory hierarchy includ[ing] one or more levels of cache within the cores, a set or one or more shared cache units 906, and external memory...coupled to the

set of integrated memory controller units 914.” *Id.* ¶94. Accordingly, “[p]rocessors 1170 and 1180 [(blue)] are shown including integrated memory controller (IMC) units 1172 and 1182 [(purple)], respectively.” *Id.* ¶106. “IMCs 1172 and 1182 couple the processors to respective memories, namely a memory 1132 and a memory 1134 [(pink)], which may be portions of main memory locally attached to the respective processors.” *Id.*

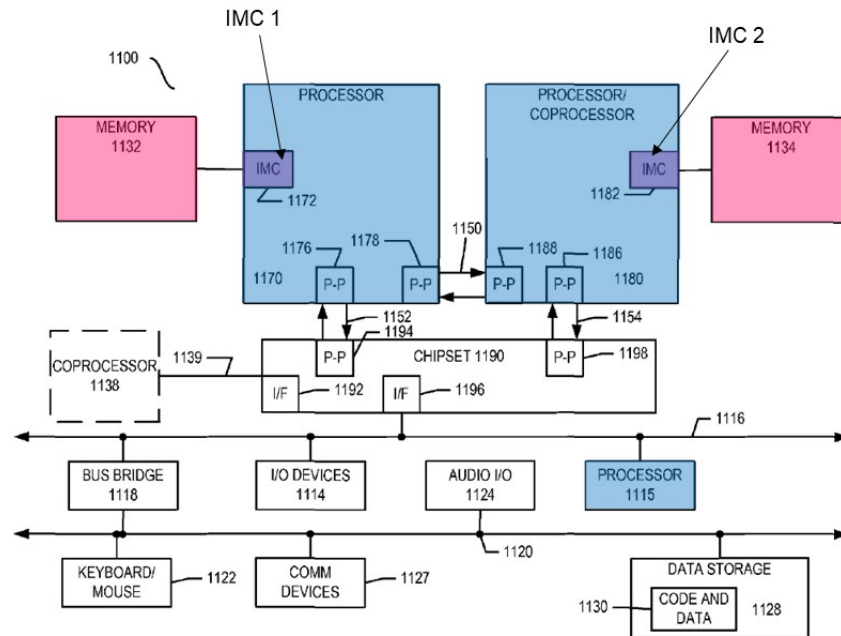


FIG. 11

Id., Fig. 11. Therefore, Moses’s IMC 1172 controls access by the symmetric, multi-threaded cores to a first system memory 1132 and IMC 1182 controls access by the symmetric, multi-threaded cores to a second system memory 1134. *See supra* Section VII.A.2.c (illustrating Moses’s teaching of symmetric, multi-threaded cores (elements 1[a]/15[a]/20[b])); *see also* Ex-1005 ¶113, Fig. 13; Ex-1002 ¶¶97, 106, 117.

3. Dependent Claims 2–4, 6, 11–14, 16–17, 21

- a. Claim 2/Claim 16: wherein, based on workload on each of the first and second clusters of the plurality of cores, the system is operative to selectively migrate a thread from the first cluster to the second cluster.**

Trautman’s DTM mechanism focuses on identifying “hot spots,” i.e., regions of increased power density that are identifiable because “the thermal temperature of the die at the regions of increased power density is greater than other regions of the die.” Ex-1007 ¶15. Moreover, “[a] higher operating power may be an indication of a greater processing function being performed by the core.” *Id.* ¶16. Trautman discloses an operation in which “at least a portion of the computational load may be distributed from the core having the highest temperature to the one core having the lowest temperature.” *Id.* ¶35. Specifically, Trautman discloses that “the basis for the routing of the computational load (e.g., an algorithm, basis, relationship, etc.) may include more, fewer, and different factors than the temperature.” *Id.* ¶33; *see also id.* ¶34.

For example, “[a] higher operating power may be an indication of a greater processing function being performed by the core.” Ex-1007 ¶16. Trautman’s Graph 120 identifies a relationship between greater processing function (i.e., workload), core temperature, and increased power density (i.e., hot spot), which justifies migrating a thread between cores to balance computational load and, thus, temperature across the die. *Id.*; Ex-1002 ¶¶118–22. Therefore, Trautman teaches

that computational workload can form the basis for migrating a thread between cores.

Further, Trautman discloses migrating a thread between core clusters based on workload on each of the clusters. Trautman explains, “the number of cores included in a cluster(s) may depend on...the computational tasks and/or computational load being processed, and other factors.” Ex-1007 ¶47. “The multiple cores may provide a power efficient device, particularly with regard to processing parallel or multi-threaded tasks.” *Id.* ¶15. Trautman’s Figure 7 (annotated) depicts migrating a thread from a first cluster to a second cluster:

Multi-core processor 725 having a plurality of cores 740 includes two clusters of cores, cluster 730 and cluster 735. For clusters 730 and 735, a DTM mechanism in accordance herewith may be invoked to dynamically distribute the processing of a computational load across the clusters (indicated by the arrows between the clusters [in multi-core processor 725]). In some embodiments, [DTM] mechanisms may be used to distribute processing between clusters 730 and 735. Here, the cores in clusters 730 and 735 may operate at a substantially higher frequency than the cores not in the cluster.

Ex-1007 ¶49.

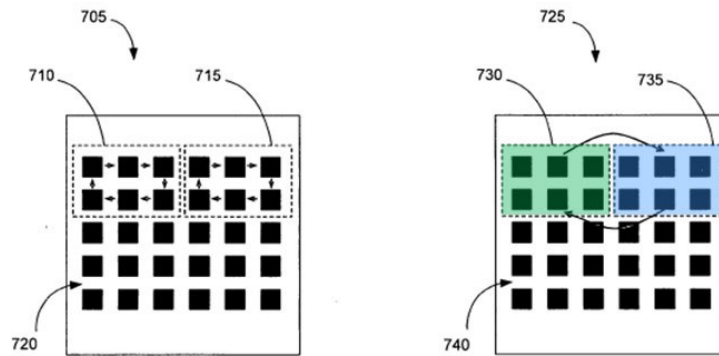


FIG. 7

Id., Fig. 7. Because the DTM mechanism is used to “distribut[e] the processing of the multi-core processor between the cores..., the effective power density on the die may be reduced through a thermal capacitance effect of the cores. The reduced power density may result in a lower die temperature.” *Id.* ¶18. Accordingly,

Trautman’s teachings of thread migration to balance computational load and, thus, temperature across the die apply equally to migration between clusters of cores.

See id. ¶35 (“[A]t least a portion of the computational load may be distributed from the core having the highest temperature to the one core having the lowest temperature.”). In addition, a POSITA would have understood or found it obvious in view of the state of the art that Trautman’s DTM mechanism discloses that computational workload can form the basis for migrating a thread between clusters of cores. Ex-1002 ¶¶118–22; *see, e.g.*, Ex-1009, code (57); *see also* Ex-1010 ¶13.

Although Moses does not disclose this limitation, a POSITA looking to

further increase Moses's system efficiency through load balancing would have looked to combine Trautman's thread migration teachings with Moses's multi-core system. Ex-1002 ¶¶118–22. For example, Trautman teaches that “[h]ot spots may present challenges to efficiently managing thermal and power dissipation aspects of the [multi-core] processor.” Ex-1007 ¶3. Trautman addresses this issue “[b]y distributing the processing of the multi-core processor between the cores.” *Id.* ¶18.

b. Claim 3/Claim 21: further comprising a Peripheral Component Interface Express (PCIe) interface coupled with the plurality of cores.

Moses discloses that “[c]hipset 1190 may be coupled to a first bus 1116 via an interface 1196,” in which “first bus 1116 may be a Peripheral Component Interconnect (PCI) bus, or a bus such as a PCI Express [(PCIe)] bus or another third generation I/O interconnect bus.” Ex-1005 ¶109; *see also id.* ¶71. Moses's Figure 11 (annotated) depicts that “various I/O devices 1114 may be coupled to first bus 1116 [(e.g., PCIe)].... In one embodiment, one or more additional processor(s) 1115...are coupled to first bus 1116.” *Id.* ¶110; Ex-1002 ¶123.

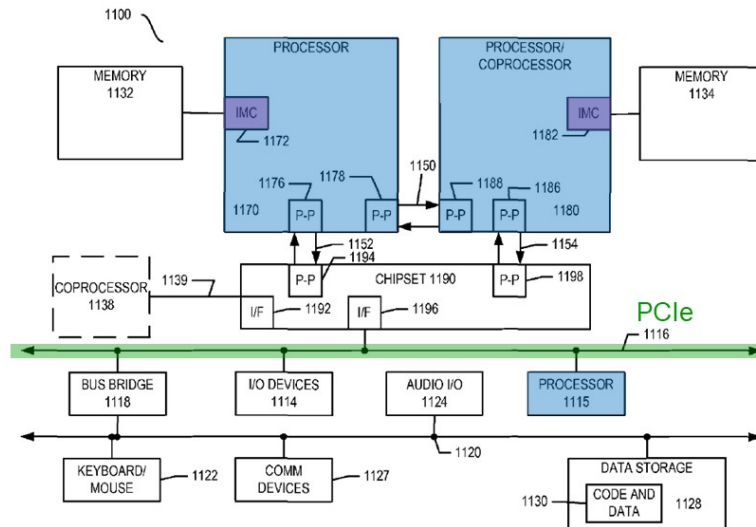


FIG. 11

Id., Fig. 11.

- c. **Claim 4/Claim 17: wherein the non-uniform access to the plurality of distributed cache portions by the plurality of cores is based, at least in part, on each distributed cache portion being coupled to a proximate cluster of the plurality of cores over a first interconnect having first characteristics and being coupled to a non-proximate cluster of the plurality of cores over a second interconnect having second characteristics.**

Because Moses discloses that “[e]ach processor core has a direct access path to its own local subset of the L2 cache 804,” which can be “accessed quickly,” Moses discloses a distributed cache portion accessing a proximate cluster of the plurality of cores over a first interconnect having first characteristics. Ex-1005 ¶89. In addition, Moses discloses that data “stored in its L2 cache subset 804...can be accessed...in parallel with other processor cores accessing their own local L2 cache subsets,” and “[t]he ring network ensures coherency for shared data. The

ring network is bidirectional to allow agents such as processor cores, L2 caches and other logic blocks to communicate with each other within the chip.” *Id.*; *see id.* ¶94; *see id.*, Fig. 9 (illustrating ring interconnect 912). Thus, Moses also discloses a ring network (i.e., a second interconnect) coupling each distributed cache portion to the non-proximate cores.

Although Moses does not explicitly disclose a first and second interconnect having first and second characteristics, a POSITA would have understood that Moses teaches non-uniform access through two interconnects with different characteristics. Ex-1002 ¶¶124–26. A POSITA would have understood that Moses’s first interconnect enables quick and direct access to a local cache portion (e.g., lower access latency) and its second interconnect provides longer access latency between the cores and the other cache portions. Ex-1005 ¶94; Ex-1002 ¶¶124–26. A POSITA would have understood that accessing a cache portion directly over a first interconnect would have lower access latency than accessing another cache portion over a ring interconnect, especially when the core accesses a cache portion that is physically far from the core. Ex-1002 ¶¶124–26; *see, e.g.*, Ex-1012 at 10, Fig. 1.8; *see also* Ex-1008 ¶¶3, 18. In addition, a POSITA would have understood that interconnects with different access latencies contributes to non-uniform cache access. Ex-1002 ¶¶124–26.

d. Claim 6

- (1) Element 6[a]: The system of claim 1, wherein based on workload on each of the first and second clusters of the plurality of cores, the system is operative to selectively migrate a thread from the first cluster to the second cluster,

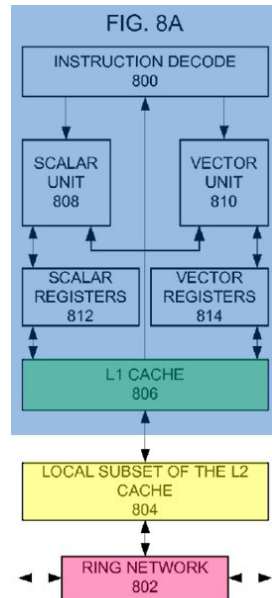
As discussed above in Section VII.A.3.a (dependent Claim 2), Moses in view of Trautman teaches element 6[a]. *See supra* Section VII.A.3.a; Ex-1002 ¶127.

- (2) Element 6[b]: and further comprising a Peripheral Component Interface Express (PCIe) interface coupled with the plurality of cores.

As discussed above in Section VII.A.3.b (dependent Claim 3), Moses discloses element 6[b]. *See supra* Section VII.A.3.b; Ex-1002 ¶128.

e. Claim 11: The system of claim 1 wherein a first core of a first predetermined cluster is physically proximate to a first distributed cache portion.

Moses discloses that “[t]he system also includes a cache 540 that is coupled to the CPU cores 510. The cache 540 is a shared cache that is shared among the CPU cores 510, and can be accessed by any of the CPU cores 510.” Ex-1005 ¶55. Further, Moses discloses that “[t]he cache 540 is organized into a plurality of cache partitions.” *Id.* ¶56. Moses’s Figure 8A (annotated) illustrates “a block diagram of a single processor core, along with its connection to the on-die interconnect network 802 and with its local subset of the Level 2 (L2) cache 804.” *Id.* ¶88.



Id., Fig. 8A. “The local subset of the L2 cache 804 is part of a global L2 cache that is divided into separate local subsets, one per processor core. Each processor core has a direct access path to its own local subset of the L2 cache 804.” *Id.* ¶89. Because the core has a direct access path to its local portion of the L2 cache and can access it quickly, a POSITA would have understood or found it obvious for Moses to couple a core and cache portion within a cluster in close physical proximity. Ex-1002 ¶¶129–30.

To the extent that Moses does not disclose a first cluster in which a first core of the cluster is physically proximate to a first cache portion, it would have been obvious over Moses in view of Trautman’s teachings of clustering. *Id.* Particularly, Trautman discloses assigning cores to groups (i.e., clusters)—based, at least in part, on a physical distance or adjacency between the cores. *See* Ex-1007 ¶¶47, 44, Fig. 6. Thus, a POSITA motivated to increase efficiency and decrease access

latency within a system would have modified Moses in view of Trautman to cluster Moses's cores, which are physically proximate to local cache portions, based on physical proximity. Ex-1002 ¶¶129–30; *see Dystar*, 464 F.3d at 1368.

- f. **Claim 12: further comprising: a local core-cache interconnect coupling the first core to the first distributed cache portion; and an on-die interconnect coupling one or more other cores to the first distributed cache portion.**

Claim 13: wherein the local core-cache interconnect and the on-die interconnect provide different access latency and/or data throughput to the first distributed cache portion.

Claim 14: wherein the local core-cache interconnect is operative to provide relatively lower latency and/or higher data throughput to the first core when the first core accesses the first distributed cache portion than the on-die interconnect is operative to provide to the one or more other cores when the one or more other cores access the first distributed cache portion.

Because Moses discloses that “[e]ach processor core has a direct access path to its own local subset of the L2 cache 804,” which can be “accessed quickly,” Moses discloses a distributed cache portion accessing a proximate cluster of the plurality of cores over a first interconnect (e.g., a core-cache interconnect). Ex-1005 ¶89. In addition, Moses discloses that data “stored in its L2 cache subset 804...can be accessed...in parallel with other processor cores accessing their own local L2 cache subsets,” and explains that “[t]he ring network is bidirectional to allow agents such as processor cores, L2 caches and other logic blocks to

communicate with each other within the chip.” *Id.* ¶¶89, 94, Fig. 9 (illustrating ring interconnect 912). Thus, Moses discloses a second interconnect (i.e., an on-die interconnect) coupling each distributed cache portion to the non-proximate cores.

Although Moses does not explicitly disclose a first and second interconnect having first and second characteristics, a POSITA would have understood that Moses teaches non-uniform access through two interconnects with different characteristics. Ex-1002 ¶¶131–33. A POSITA would have understood that the processor cores have a first interconnect enabling quick and direct access to a cache portion (e.g., a local core-cache interconnect with lower access latency) and a second interconnect (e.g., an on-die interconnect providing longer access latency) that enables communication between the cores and the other cache portions. *See* Ex-1005 ¶94; Ex-1002 ¶¶131–33. A POSITA would have understood that accessing a cache portion directly over a first interconnect would have lower access latency than accessing another cache portion over a ring interconnect, especially when the core accesses a cache portion that is physically far from the core. Ex-1002 ¶¶131–33; *see, e.g.*, Ex-1008 ¶¶3, 18.

To the extent that Moses does not explicitly disclose a first and second interconnect with independent characteristics connecting the cache portions to proximate and non-proximate clusters, it would have been obvious to modify Moses in view of the knowledge of a POSITA. Ex-1002 ¶¶131–33. For example, a

POSITA would have understood that “[t]he banks of [a] NUCA cache...may be arranged according to a distance hierarchy with respect to [its] core[s]” and that “[t]he distance hierarchy may refer to the several levels of delay or access time.” Ex-1006 ¶31; Ex-1002 ¶¶131–33. A POSITA would have understood that “caches may contain memory banks that are connected with wires, traces, or interconnections,” which “introduce various delays.” Ex-1006 ¶¶41, 31; Ex-1002 ¶¶131–33. Thus, a POSITA would have been motivated to modify Moses’s interconnects to include a first and second interconnect with different access latency and/or data throughputs because the purpose of caching is to increase data retrieval performance. Ex-1002 ¶¶131–33; *see Dystar*, 464 F.3d at 1368.

B. Ground 2: Claims 1–3, 6, 11, 15, 16, 20, 21 Are Obvious Over Balakrishnan in View of Trautman

The subject matter of the challenged claims is disclosed by Balakrishnan alone or in view of other secondary references. For example, Balakrishnan’s Figures 1, 2A, and 4D depict most of the limitations of the challenged claims “in which like references may indicate similar elements.” *See* Ex-1006 ¶¶13–15, 20. Thus, a POSITA would understand that the systems depicted in Balakrishnan’s Figures 1, 2A, and 4D represent the same system or at least compatible systems with similar elements. Ex-1002 ¶134.

**1. A POSITA would have been motivated to combine
Balakrishnan with Trautman**

A POSITA would have been motivated to combine Balakrishnan and Trautman and would have a reasonable expectation of success in doing so, because they are both from the same field and relate to the same well-known issues. Ex-1002 ¶135. Balakrishnan and Trautman provide complementary solutions for power management of multi-core processors. *Compare* Ex-1006 ¶¶1, 9–12, *with* Ex-1007 ¶¶15–18. A POSITA seeking to increase system efficiency in a multi-core processor would have known of and consulted these proposed solutions. Ex-1002 ¶135.

Balakrishnan discloses techniques for power conservation in non-uniform cache access systems. Ex-1006 ¶¶7, 9–12. Trautman provides additional well-known teachings of dynamic thermal management to reduce leakage power of a processor. Ex-1007 ¶18. A POSITA would have been motivated to modify Balakrishnan to obtain additional features and benefits, such as multi-threaded cores, decreased power leakage through consideration of thermal temperature and power density, and improved system power management through clustering. Ex-1002 ¶136; *see* Ex-1007 ¶¶15, 18, 58.

A POSITA would have had a reasonable expectation of success in combining the teachings of Balakrishnan and Trautman with no change in their respective functions because both references purport to solve these problems in

similar and predictable ways, such as using well-known techniques, including multi-threading, thread migration, and clustering. Ex-1002 ¶137.

2. Independent Claims 1, 15, 20⁴

a. Element 1[preamble]

Element 20[preamble]

Element 15[preamble]

Balakrishnan is directed to a machine or system that “conserv[es] power in non-uniform cache access (NUCA) systems.” Ex-1006 ¶1; *see also id.* ¶¶11, 28, Fig. 1. “Some embodiments comprise a method that includes enabling a number of banks of a NUCA cache, with the ways of the cache being vertically distributed across multiple banks.” *Id.* ¶9; *see also id.* ¶6, Fig. 6. Balakrishnan also teaches that “[o]ne or more elements of the apparatuses in FIGS. 4A–4D may be in the form of hardware [(i.e., machines, systems)], software [(i.e., methods)], or a combination of both hardware and software.” *Id.* ¶54; *see also id.* ¶¶64, 81; Ex-1002 ¶¶138, 159, 169.

b. Element 20[a]

Balakrishnan discloses that “[s]ystem 100 has four processors, 105, 110, 115, and 120.... Each processor may comprise one or more cores.” Ex-1006 ¶29,

⁴ Although the claim language of independent Claims 1, 15, and 20 differ slightly, the claim requirements are substantively identical and any meaningful differences are analyzed.

Fig. 1; Ex-1002 ¶170.

c. Elements 1[a]/15[a]/20[b]

First, Balakrishnan discloses executing instructions and processing data with a plurality of cores: “The instructions, when executed by at least one processor, may also sequentially turn off groups of banks.” Ex-1006 ¶¶12, 36, 64; *see id.* ¶24 (“[A] cache may have many blocks which individually store the various instructions and data values.”); Ex-1002 ¶¶139–43, 160–61, 171.

Second, Balakrishnan in view of Trautman teaches a plurality of symmetric, multi-threaded cores. Balakrishnan discloses that “[d]ifferent embodiments may comprise different numbers of processors,” and “[e]ach processor may comprise one or more cores.” Ex-1006 ¶29. Although Balakrishnan does not explicitly describe the cores as symmetric cores, a POSITA would have found it obvious to implement Balakrishnan with symmetric cores because there were only two options: symmetric or asymmetric cores. Ex-1002 ¶¶139–43, 160–61, 171. In addition, symmetric cores were well-known and the obvious choice for ease of implementation and efficiency. *Id.*; *see, e.g.*, Ex-1011 at 447. Balakrishnan’s Figure 1 depicts several processors (i.e., 105–20)—including processor 105 with cores 125 and 130—coupled to system memory 175 and peripherals 165. *Id.*, Fig. 1. Because Cores 105–20 are represented identically and treated equally, a POSITA would have understood that Balakrishnan’s Figure 1 illustrates a system

with symmetric cores. Ex-1002 ¶¶139–43, 160–61, 171; *see* Ex-1006 ¶¶29–30, 40.

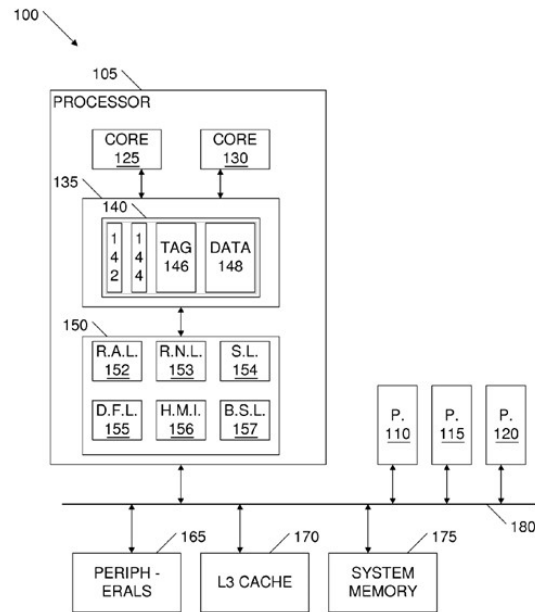


FIG. 1

Id., Fig. 1.

A POSITA would have found it obvious to implement Balakrishnan’s multi-core processors with multi-threaded cores because there were only two options: single-threaded or multi-threaded cores. Ex-1002 ¶¶139–43, 160–61, 171. Because multi-threaded cores can run threads in parallel, a system with multi-threaded cores would handle a context-switching workload faster than a system with single-threaded cores. *Id.*; *see, e.g.*, Ex-1011 at 446. If a POSITA decided to use single-threaded cores, they would need more cores to handle the same workload in the same amount of time. Ex-1002 ¶¶139–43, 160–61, 171. In addition, multi-threaded cores were well-known and would have been an obvious consideration when

designing Balakrishnan's system. *Id.* Thus, it would be obvious to a POSITA to use multi-threaded cores in a multi-core system such as Balakrishnan's Figure 1 to keep up with the workload of multiple multi-threaded applications. Ex-1002 ¶¶139–43, 160–61, 171; *see Dystar*, 464 F.3d 1356 at 1368.

Further, Trautman teaches that “[a] device, system, platform, or operating environment may include more than one processor or a processor having more than one core (i.e., a multi-core processor).” Ex-1007 ¶2. Trautman states:

The security, reliability, and efficient operation of such a device...may be enhanced by the inclusion and use of the multi-core processor. For example, a multi-core processor may provide the processing performance of multiple processors by executing multiple threads of instruction in parallel while consuming less power, costing less, and using less space than multiple single-core processors.

Id. “The multiple cores may provide a power efficient device, particularly with regard to processing parallel or multithreaded tasks.” *Id.* ¶15. Accordingly, a POSITA would have been motivated to combine Balakrishnan with Trautman's teachings of multi-threaded cores to increase system efficiency. Ex-1002 ¶¶139–43, 160–61, 171.

d. Elements 1[b]/15[b]/20[c]

Balakrishnan discloses, “one or more of the L1 cache and L2 cache structures [of system 100], as well as L3 cache 170, may comprise NUCA caches.”

Ex-1006 ¶30. Balakrishnan further discloses that “a cache may have many blocks which individually store the various instructions and data values.” *Id.* ¶24; *id.* ¶25 (“An exemplary cache line (block) may include an address tag field, a state bit field, an inclusivity bit field, and a value field for storing the actual instruction or data.”). Ex-1002 ¶¶144, 162, 172.

e. Elements 1[c]/15[c]/20[d]

Balakrishnan’s Figure 1 (annotated) “depicts a system 100 arranged to conserve power in a vertically distributed NUCA cache 135.” Ex-1006 ¶28. Balakrishnan discloses that “[w]hile not specifically depicted in FIG. 1, cores 125 and 130 [(blue)] may also each comprise L1 cache [(green)].” *Id.* ¶30; *see also id.* ¶44 (“While not shown in FIG. 2A, each of processors [260–66] may comprise small single-banked caches, e.g., L1 caches.”). Because the L1 cache is private to each core, a POSITA would have understood or found it obvious that Balakrishnan discloses at least one first level cache integral to one of the plurality of cores. Ex-1002 ¶¶145–47, 163, 173.

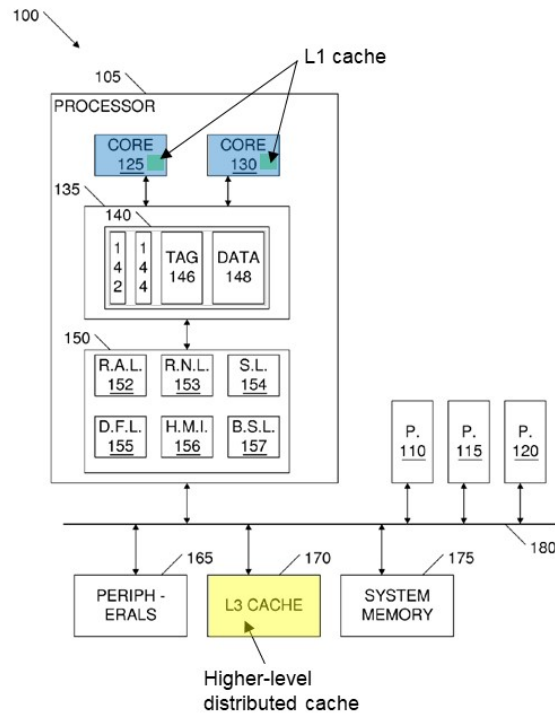


FIG. 1

Id., Fig. 1.

In addition, Balakrishnan explains:

System 100 may also have one or more L2 cache elements, such as NUCA cache 135. NUCA cache 135 may comprise a number of banks, such as bank 140.... In various embodiments of system 100, one or more of the L1 cache and L2 cache structures, as well as L3 cache 170, may comprise NUCA caches.

Ex-1006 ¶30. Further, Balakrishnan explains that “researchers have begun proposing designs that integrate a very large (e.g., multiple megabytes) third level (L3) cache memory [(yellow)] on the same die as the processor core for improved data processing performance.” *Id.* ¶4. “Large on-die cache memories are typically subdivided into multiple cache memory banks, which are then coupled to a

wide...data bus.” *Id.* ¶5; *see id.* Fig. 2A (showing cache banks 200–33 distributed across a die). Therefore, Balakrishnan discloses at least one higher level distributed cache comprising a plurality of distributed cache portions that are physically distributed across a die.

Further, as shown in Figure 2A, Balakrishnan discloses that each distributed cache portion is accessible to, and therefore shared by, each of the plurality of cores. Balakrishnan states:

System 270 may provide static power savings in vertically-striped multibank cache 240 by taking processor-bank distance/latency into account. For each bank in cache 240 a relative distance may be computed. In computing the relative distances for the banks of cache 240, one may tally the horizontal and vertical distances from bank-to-bank, elements 250 and 252, respectively, as well as the bank-to-processor distance(s), element 254. The relative distance for a bank may be defined as the sum of the distances from the bank to each of the processors.

Ex-1006 ¶47.

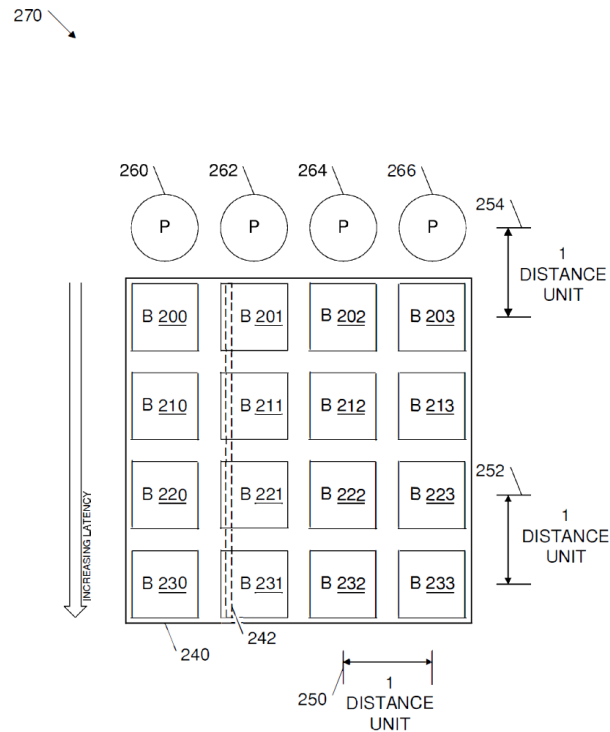


FIG. 2A

Id., Fig. 2A; ¶44 (“While not shown in FIG. 2A, each of processors 260, 262, 264, and 266 may comprise small single-banked caches, e.g., L1 caches. Processors 260, 262, 264, and 266 may also be coupled with large multi-banked lower level cache 240. Cache 240 may comprise a NUCA cache.”). Because Balakrishnan takes into account the “bank-to-processor” distance for each bank in cache 240, a POSITA would understand that each processor may access each of the banks in cache 240. Ex-1002 ¶¶145–47, 163, 173. For example, Balakrishnan’s Table 1 contemplates the bank-to-processor distance among each of the four processors and all sixteen banks of Figure 2A. Ex-1006, Table 1. Further, Balakrishnan explains that “one or more of the processors [260, 262, 264, and 266] may be replaced by cores,” so a POSITA would understand that each core of this system

may access each of the banks in cache 240. *Id.* ¶44; Ex-1002 ¶¶145–47, 163, 173.

f. Elements 1[d]/15[d]/20[e]

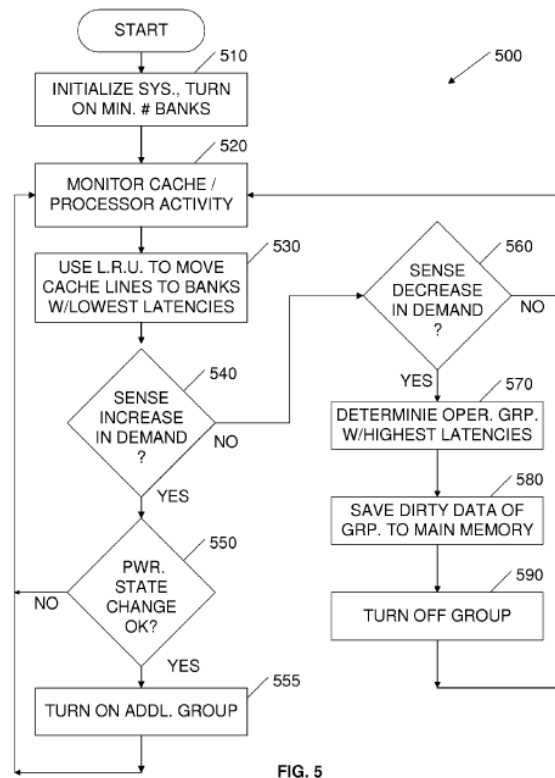
Balakrishnan discloses that “the caches of system 100 may be coherent and utilize a coherency protocol.... Each cache level, from highest (L1) to lowest (L3), may successively store more information, but at a longer access penalty.” Ex-1006 ¶40. Balakrishnan specifically discloses cache management circuitry such as “[c]ache controller 150” to “control NUCA cache 135 by using various cache operations. These cache operations may include placement, eviction or replacement, filling, coherence management, etc.” *Id.* ¶38. Further, Balakrishnan explains that “one or more of the L1 cache and L2 cache structures, as well as L3 cache 170, may comprise NUCA [Non-Uniform Cache Access/Architecture] caches.” *Id.* ¶30; *see id.* ¶7. “In NUCA caches, the latency to a bank generally depends on the proximity to the device making the request, which frequently is a processor. NUCA allows banks closest to the processor to respond the fastest and forces the banks furthest from the processor to respond the slowest.” *Id.* ¶¶7, 38; *see id.* ¶31 (“NUCA cache 135 may store data and associated tags in a non-uniform access manner.”). Ex-1002 ¶¶148, 164, 174.

g. Element 20[f]

Balakrishnan discloses, “a computer program product of a computer readable storage medium including instructions.... The instructions, when

executed by at least one processor, may also sequentially turn off groups of banks.”

Ex-1006 ¶12. Balakrishnan’s Figure 5 “depicts a flowchart 500 illustrating how an embodiment may operate a plurality of banks of a NUCA cache, sense increases or decreases in demand, and switch banks of the NUCA cache accordingly.” *Id.* ¶64.



Id., Fig. 5. “For example, the embodiment may be implemented as a computer program product comprising a computer readable storage medium including instructions that, when executed by a processor conserve power in the system depicted in FIG. 1.” *Id.* ¶64; Ex-1002 ¶175.

h. Elements 1[e]/15[e]/20[g]

Trautman’s Figure 6 (annotated) illustrates:

a multi-core processor 615 [misabeled 625] including an array of cores 630 wherein a number of cores 630 are grouped into two clusters 620 [(green)] and 625 [(blue)]. Clusters 620 and 625 may, under the control of a DTM mechanism in accordance herewith, operate as superscalar clusters that are selected from adjacent cores 630. Clusters 620 and 625 may operate at a higher frequency than the remaining cores of multi-core processor 615 not included in clusters 620 and 625.

Ex-1007 ¶44. Further, Trautman explains that “the number of cores included in a cluster may vary,” “[t]he clustered cores may be adjoining,” and “the configuration or groupings of cores may be predetermined or vary in accordance with operational contexts.” *Id.* ¶47. “For example, the number of cores included in a cluster(s) may depend on the number of cores available for clustering, the power to be dissipated, the computational tasks and/or computational load being processed, and other factors.” *Id.*

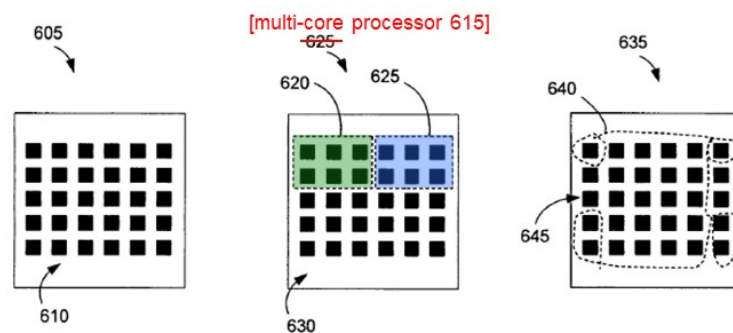


FIG. 6

Id., Fig. 6. A POSITA would have understood that clustering adjoining or

physically proximate cores would decrease access latency within the cluster and allow for frequency control or power gating at the cluster level. Ex-1002 ¶¶149, 165, 176. Thus, a person motivated to increase system efficiency and ease of operation within a system would have modified Moses in view of Trautman's teachings of clustering and frequency control at the cluster level. *Id.*

i. Elements 1[f]/15[f]/20[h]

Trautman discloses that “a cluster may include at least two cores,” which “may be adjoining.” Ex-1007 ¶47; *see id.* ¶45. For example, Trautman's Figure 6 depicts “a multi-core processor 615 including an array of cores 630 wherein a number of cores 630 are grouped into two clusters 620 and 625. Clusters 620 and 625 may...operate as superscalar clusters that are selected from adjacent cores.” *Id.* ¶44.

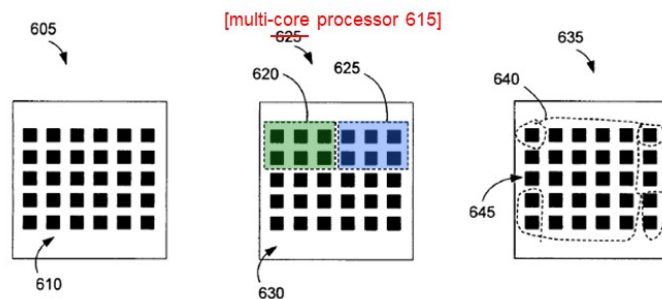


FIG. 6

Id., Fig. 6. As shown in Figure 6, clusters 620 and 625 comprise physically proximate cores. *See id.* Accordingly, a POSITA would have understood that the

average distance between cores in either cluster 620 or 625 was less than the average distance between all of the cores of the array 630. Ex-1002 ¶¶150–51, 166, 177.

Further, Balakrishnan discloses that “[i]n NUCA caches, the latency to a bank generally depends on the proximity to the device making the request, which frequently is a processor. NUCA allows banks closest to the processor to respond the fastest and forces the banks furthest from the processor to respond the slowest.” Ex-1006 ¶7; *see id.* ¶¶31, 38. Accordingly, a POSITA would have understood that clustering adjoining or physically proximate cores to achieve a lower average distance between cores within a cluster would decrease access latency and allow for frequency control or power gating at the cluster level, which was well-known in the art. Ex-1002 ¶¶150–51, 166, 177; *see, e.g.*, Ex-1009 ¶22. Therefore, a POSITA would have been motivated to combine Balakrishnan’s teachings of access latency within a cache and Trautman’s teachings of core clustering to increase system efficiency while maintaining low access latency between the cores and cache portions within a cluster. Ex-1002 ¶¶150–51, 166, 177.

j. Elements 1[g]/15[g]/20[i]

Balakrishnan’s Figure 4D (annotated) “depicts...apparatus 470 configured to conserve power in NUCA caches [480–86]. Apparatus 470 comprises a power controller 476 coupled to processors 474 and 472, as well as NUCA caches [480–

86].” Ex-1006 ¶62. Because Balakrishnan explains that “[e]ach processor may comprise one or more cores,” processors 474 and 472 represent a first and second cluster of cores. *Id.* ¶29 (“For example, processor 105 comprises two cores, 125 and 130, in the embodiment depicted in FIG. 1.”); *see also id.* ¶44 (“[D]epending on the embodiment and/or technology, one or more of the processors may be replaced by cores.”). “[P]ower controller 476 may turn off and on various banks in NUCA caches [480–86] to switch between a number of different power states. Upon turning off an entire NUCA cache element, power controller 476 may conserve additional power by turning off a core of a processor or an entire processor.” *Id.* ¶62.

Balakrishnan discloses:

[P]ower controller 476 [(red)] may sequentially turn off groups of banks of NUCA cache 480 [(yellow)] until all of the banks of NUCA cache 480 have been turned off. Upon recognizing that all of the banks of NUCA cache 480 have been turned off, power controller 476 may turn off a core in processor 472 [(i.e., cluster of cores) (blue)] associated with NUCA cache 480.

Ex-1006 ¶62 (explaining that processor 472 (blue) and corresponding NUCA caches 480 (yellow) and 482 (purple) can be turned off together, and similarly, processor 474 (green) and corresponding NUCA caches 484 (teal) and 486 (orange) can be turned off together).

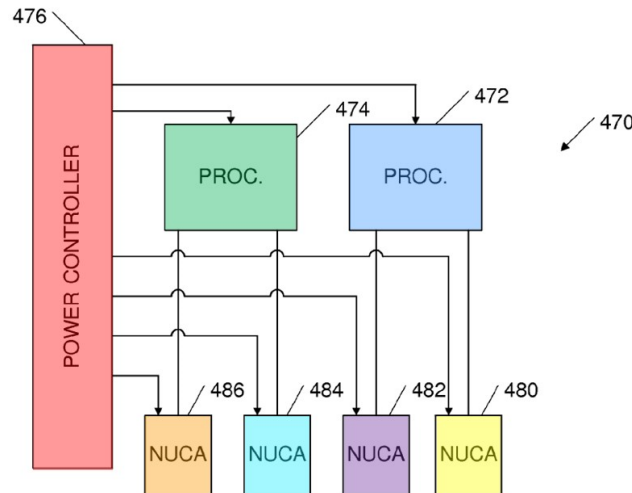


FIG. 4D

Id., Fig. 4D. In addition, Balakrishnan discloses that an “activity monitor may detect when an apparatus or system enters an idle state or a low activity state for an extended period of time” and “automatically trigger the state selector to turn off groups of banks, based on the monitored activity, to conserve power” or “turn on additional groups of banks to meet the demand of the increased activity.” *Id.* ¶63.

Further, Balakrishnan discloses that “[d]ifferent embodiments may turn off different amounts of NUCA L1, L2, and L3 caches to conserve power.” Ex-1006 ¶40. Balakrishnan discloses “banks of a non-uniform cache access (NUCA) cache, with the ways being vertically distributed across multiple banks,” in which “[b]anks may generally be assigned to different groups based on access latencies.” *Id.* ¶10. Balakrishnan’s embodiments “comprise switches configured to turn off groups of banks” and “state selectors...coupled to the switches to select different power states of the NUCA caches.” *Id.* “The state selectors are arranged to turn off groups of banks with the greatest access latencies before turning off other groups

with smaller latencies.” *Id.* Thus, a POSITA would have understood that Balakrishnan discloses gating power to at least one cluster of cores and cache portions or, alternatively, would have been motivated to modify Balakrishnan to teach clustering the cores and cache portions to gate power at the cluster level. Ex-1002 ¶¶152–54, 167, 178.

k. Elements 1[h]/15[h]/20[j]

Balakrishnan discloses that its system may include either “a memory controller hub (MCH)...or an integrated memory and I/O controller.” Ex-1006 ¶42. Either of these components would serve as an integrated memory controller. Ex-1002 ¶¶155–58. In addition, Balakrishnan discloses “System memory 175,” which “may comprise dynamic random access memory (DRAM) in many embodiments.” Ex-1006 ¶34. Accordingly, Balakrishnan at least teaches a first integrated memory controller controlling access to a first system memory. Ex-1002 ¶¶155–58, 168, 179.

A POSITA would have been motivated to modify Balakrishnan to include a second integrated memory controller and system memory because including an additional system memory would increase the memory bandwidth of a system by allowing the memories to operate in parallel. Ex-1002 ¶¶155–58, 168, 179. Further, a POSITA would have known that using multiple DRAM memories would require an integrated memory controller for each DRAM. *Id.*

To the extent that Balakrishnan alone does not teach a first and second integrated memory controller and first and second system memory, Balakrishnan in view of Trautman teaches this limitation. Ex-1002 ¶¶155–58, 168, 179. Trautman’s Figure 10 depicts “a system 1000 including an apparatus, for example a multi-core processor 1005 in communication with a controller 1010. A memory 1015 is attached to controller 1010 by a conductor and other electrical connections.” Ex-1007 ¶63.

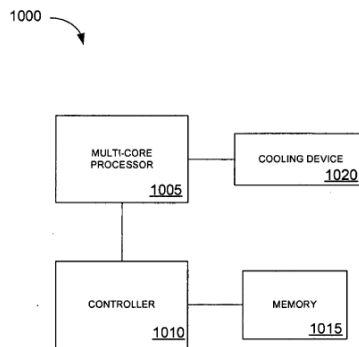


FIG. 10

Id., Fig. 10. Trautman explains that “[t]hose in the art should appreciate that system 1000 may include additional, fewer, or alternative components to multi-core processor 1005, controller 1010, memory 1015, and cooling device 1020.” *Id.* ¶65; *see id.* ¶¶32, 41. Because controller 1010 is coupled with memory 1015 and controls access by the multi-core processor—comprising symmetric, multi-threaded cores—to memory 1015, a POSITA would have understood that controller 1010 is an integrated memory controller. Ex-1002 ¶¶155–58, 168, 179.

A POSITA would have been motivated to include a second integrated memory controller and system memory because doing so would increase the memory bandwidth of the system. *Id.*

3. Dependent Claims 2, 3, 6, 7, 9, 11, 16, 18, 21

a. Claim 2/Claim 16

Trautman's DTM mechanism focuses on identifying "hot spots," i.e., regions of increased power density that are identifiable because "the thermal temperature of the die at the regions of increased power density is greater than other regions of the die." Ex-1007 ¶¶15. Moreover, "[a] higher operating power may be an indication of a greater processing function being performed by the core." *Id.* ¶16. Trautman discloses an operation in which "at least a portion of the computational load may be distributed from the core having the highest temperature to the one core having the lowest temperature. The basis for the routing may vary in accordance with the various embodiments herein." *Id.* ¶35. Specifically, Trautman discloses that "the basis for the routing of the computational load (e.g., an algorithm, basis, relationship, etc.) may include more, fewer, and different factors than the temperature." *Id.* ¶¶33, 34.

For example, "[a] higher operating power may be an indication of a greater processing function being performed by the core." Ex-1007 ¶16. Trautman's Graph 120 identifies a relationship between greater processing function (i.e., workload),

core temperature, and increased power density (i.e., hot spot), which justifies migrating a thread between cores to balance computational load and, thus, temperature across the die. *See id.* ¶16; Ex-1002 ¶¶180–84. Therefore, Trautman teaches that computational workload can form the basis for migrating a thread between cores.

Further, Trautman teaches migrating a thread between core clusters based on workload on each of the clusters. Trautman explains, “the number of cores included in a cluster(s) may depend on...the computational tasks and/or computational load being processed, and other factors.” Ex-1007 ¶47. “The multiple cores may provide a power efficient device, particularly with regard to processing parallel or multi-threaded tasks.” *Id.* ¶15. Trautman’s Figure 7 (annotated) depicts selectively migrating a thread from a first cluster to a second cluster:

Multi-core processor 725 having a plurality of cores 740 includes two clusters of cores, cluster 730 and cluster 735. For clusters 730 and 735, a DTM mechanism in accordance herewith may be invoked to dynamically distribute the processing of a computational load across the clusters (indicated by the arrows between the clusters [in multi-core processor 725]).... [DTM] mechanisms may be used to distribute processing between clusters 730 and 735.

Id. ¶49.

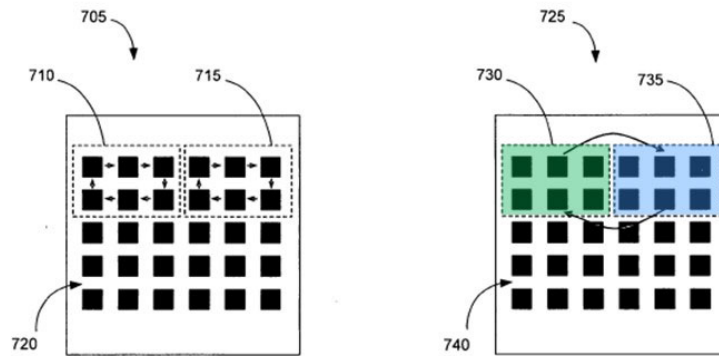


FIG. 7

Id., Fig. 7. Because the DTM mechanism is used to “distribut[e] the processing of the multi-core processor between the cores..., the effective power density on the die may be reduced through a thermal capacitance effect of the cores. The reduced power density may result in a lower die temperature.” *Id.* ¶18. Accordingly,

Trautman’s teachings of thread migration to balance computational load and, thus, temperature across the die apply equally to migration between clusters of cores.

See id. ¶35 (“[A]t least a portion of the computational load may be distributed from the core having the highest temperature to the one core having the lowest temperature.”); Ex-1002 ¶¶180–84. In addition, a POSITA would have understood or found it obvious in view of the state of the art that Trautman’s DTM mechanism discloses that computational workload can form the basis for migrating a thread between clusters of cores. *Id.*; *see, e.g.*, Ex-1009, code (57); *see also* Ex-1010 ¶13.

Although Balakrishnan does not disclose this limitation, a POSITA looking

to further increase Balakrishnan's system efficiency, would have looked to combine Trautman's teachings of thread migration between clusters with Balakrishnan's multi-core system. Ex-1002 ¶¶180–84. For example, Trautman teaches that “[h]ot spots may present challenges to efficiently managing thermal and power dissipation aspects of the processor,” and specifically, “a multi-core processor may have a tendency to have a greater number or intensity of hot spots as compared to a single core processor.” Ex-1007 ¶3. Trautman addresses this issue “[b]y distributing the processing of the multi-core processor between the cores.” *Id.* ¶18.

b. Claim 3/Claim 21

Balakrishnan discloses:

[P]eripherals 165 may be connected to a fabric bus 180 via, e.g., a peripheral component interconnect (PCI) local bus using a PCI host bridge. A PCI bridge may provide a low latency path through which processors...may access PCI devices mapped within bus memory or I/O address spaces. A PCI host bridge may also provide a high bandwidth path to allow the PCI devices to system memory 175.

Ex-1006 ¶43. Although PCI is a parallel interface and PCIe is a serial interface, PCI and PCIe are software-compatible and perform the same function. Ex-1002 ¶185; Ex-1030 at 27–29. Further, a POSITA would have been motivated to modify Balakrishnan to use a PCIe rather than a PCI because a PCIe has a significantly

higher potential bandwidth than the older PCI. Ex-1002 ¶185; Ex-1030 at 27–29.

Thus, a POSITA would have found it obvious to modify Balakrishnan to use a PCIe rather than a PCI. Ex-1002 ¶185.

c. Claim 6

(1) Element 6[a]

As discussed above in Section VII.C.3.a (dependent Claim 2), Balakrishnan in view of Trautman teaches element 6[a]. *See supra* Section VII.C.3.a; Ex-1002 ¶186.

(2) Element 6[b]

As discussed above in Section VII.C.3.b (dependent Claim 3), Balakrishnan teaches element 6[b]. *See supra* Section VII.C.3.b; Ex-1002 ¶187.

d. Claim 11

Balakrishnan discloses that “[t]he banks of NUCA cache 135 may be arranged according to a distance hierarchy with respect to [its] core[s]” and that “[t]he distance hierarchy may refer to the several levels of delay or access time.” Ex-1006 ¶31. Moreover, Balakrishnan discloses that “[b]anks may generally be assigned to different groups based on access latencies.” *Id.* ¶10. Thus, a POSITA would have understood Balakrishnan to teach non-uniform cache access and been motivated to further reduce access latencies by coupling a cluster with a proximate cache portion. Ex-1002 ¶¶188–89.

To the extent that Balakrishnan does not explicitly disclose a first cluster in which a first core of the cluster is physically proximate to a first cache portion, it would have been obvious over Balakrishnan in view of Trautman's teachings of clustering. Ex-1002 ¶¶188–89. Particularly, Trautman teaches assigning cores to groups (i.e., clusters)—based, at least in part, on a physical distance or adjacency between the cores. *See* Ex-1007 ¶¶47, 44, Fig. 6. Thus, a person motivated to increase efficiency and decrease access latency within a system would have modified Balakrishnan in view of Trautman to cluster cores—and particularly couple cores and cache portions—based on physical proximity. Ex-1002 ¶¶188–89.

C. Ground 3: Claims 4, 11–14, 17 Are Obvious Over Balakrishnan in View of Trautman and Hughes

1. A POSITA would have been motivated to combine Balakrishnan with Trautman and Hughes

A POSITA would have been motivated to combine Balakrishnan, Trautman and Hughes, and would have a reasonable expectation of success in doing so, because they are from the same field and relate to the same well-known issues. Ex-1002 ¶190. Balakrishnan and Trautman provide complementary solutions for power management of multi-core processors, and Balakrishnan and Hughes provide complementary solutions for managing a shared upper-level cache in a non-uniform cache access system. *Compare* Ex-1006 ¶¶1, 9–12, *with* Ex-1007 ¶¶15–18, *and* Ex-1008 ¶¶2, 16–19. A POSITA seeking to increase system

efficiency in a multi-core processor would have known of and consulted these proposed solutions. Ex-1002 ¶190.

Balakrishnan discloses techniques for power conservation in non-uniform cache access systems. Ex-1006 ¶¶7, 9–12. Trautman provides additional well-known teachings of dynamic thermal management to reduce leakage power of a processor, Ex-1007 ¶18, and Hughes provides well-known teachings of implementing a shared upper-level cache in a non-uniform cache access system. Ex-1008 ¶¶2, 16–19. A POSITA would have been motivated to modify Balakrishnan to obtain additional features and benefits, such as decreased access latency through separate interconnections, decreased power leakage through consideration of thermal temperature and power density, and improved system power management through clustering. Ex-1002 ¶191; *see* Ex-1007 ¶¶15, 18, 58; *see also* Ex-1008 ¶¶2, 16–19.

A POSITA would have had a reasonable expectation of success in combining the teachings of Balakrishnan, Trautman, and Hughes with no change in their respective functions because each purports to solve these problems in similar and predictable ways. Ex-1002 ¶192. For example, by using well-known implementations—such as NUCA and interconnects with varying characteristics—and techniques—such as thread migration and clustering. *Id.*

2. Claim 4/Claim 17

Balakrishnan discloses that “[t]he banks of NUCA cache 135 may be arranged according to a distance hierarchy with respect to [its] core[s]” and that “[t]he distance hierarchy may refer to the several levels of delay or access time.” Ex-1006 ¶31. Further, Balakrishnan discloses that “caches may contain memory banks that are connected with wires, traces, or interconnections,” which “introduce various delays.” *Id.* 41; *see id.* ¶31. A POSITA would have understood Balakrishnan to teach non-uniform cache access and been motivated to further reduce access latencies by using different interconnects with differing characteristics. Ex-1002 ¶¶193–96.

Moreover, Balakrishnan discloses that “[b]anks may generally be assigned to different groups based on access latencies.” Ex-1006 ¶10. A POSITA would have been motivated to implement a first interconnect with first characteristics (e.g., a local core-cache interconnect with lower access latency and higher data throughput) to couple the cores of a proximate cluster and their local cache portions while maintaining access to each cache portion by all of the non-proximate clusters over a second interconnect with second characteristics (e.g., an on-die interconnect with different access latency and/or data throughput). Ex-1002 ¶¶193–96; *see* Ex-1012 at 10–12, Fig. 1.8; *see also* Ex-1008 ¶¶3, 18. In addition, a POSITA would have understood that the different access latencies offered by these

interconnects contribute to providing non-uniform access to the plurality of distributed cache portions. Ex-1002 ¶¶193–96.

To the extent that Balakrishnan does not explicitly teach a first and second interconnect, Balakrishnan in view of Hughes renders this limitation obvious. *Id.* Hughes’s Figure 5 (annotated) depicts “a non-uniform cache architecture,” in which “a number of cache molecules 510–518 [(yellow)] and processor cores 520–528 [(blue)] may be interconnected with a dual ring interconnect [(i.e., a first interconnect) (purple)], having a clockwise ring 552 and a counter-clockwise ring 550.” Ex-1008 ¶34.

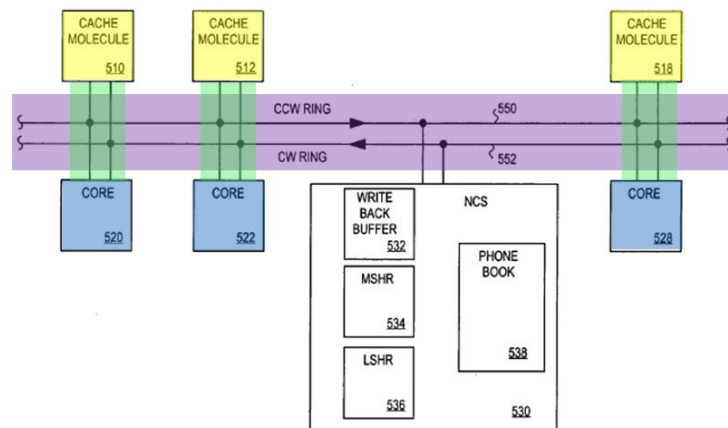


FIG. 5

Id., Fig. 5.

In addition, Figure 5 illustrates how each core is paired with a cache molecule (i.e., a cache portion) over a local interconnect (i.e., a second interconnect) (green). Ex-1008 ¶18. “The pairing is to logically associate a core

with the ‘closest’ cache molecule in terms of low access latency.” *Id.* ¶18, code (57) (“Each of the cache molecules may be closest, in terms of access latency time, to one of the processor cores.”). A POSITA would have understood the local core-cache interconnect to provide direct access to the cache molecule by its associated core and, thus, lower latency access than the ring interconnect. Ex-1002 ¶¶193–96; Ex-1012 at 10–12, Fig. 1.8. Accordingly, a POSITA would have understood Hughes to teach a first and second interconnect with first and second characteristics (e.g., access latency and/or data throughput), contributing to non-uniform cache access. Ex-1002 ¶¶193–96. Therefore, a POSITA would have been motivated to further reduce access latencies of Balakrishnan by using separate interconnects to couple cache portions with proximate and non-proximate clusters of cores. *Id.*

3. Claim 11

Balakrishnan discloses that “[t]he banks of NUCA cache 135 may be arranged according to a distance hierarchy with respect to [its] core[s]” and that “[t]he distance hierarchy may refer to the several levels of delay or access time.” Ex-1006 ¶31. Moreover, Balakrishnan discloses that “[b]anks may generally be assigned to different groups based on access latencies.” *Id.* ¶10. Thus, a POSITA would have understood Balakrishnan to disclose non-uniform cache access and been motivated to further reduce access latencies by coupling a cluster with a

proximate cache portion. Ex-1002 ¶¶197–98.

To the extent that Balakrishnan does not explicitly disclose a first cluster in which a first core of the cluster is physically proximate to a first cache portion, it would have been obvious over Balakrishnan in view of Hughes’s teachings of clustering. Ex-1002 ¶¶197–98. Particularly, Hughes teaches that “[e]ach core of cores 102–116 is...paired with a cache molecule of cache molecules 120–134. The pairing is to logically associate a core with the ‘closest’ cache molecule in terms of low access latency.” Ex-1008 ¶18; *see id.* ¶19 (“A metric of ‘distance’ may be used to describe a latency ordering of cache molecules with respect to a particular core,” which “may correlate to a physical distance between the core and the cache molecule along the interconnect.”). Thus, a POSITA motivated to increase efficiency and decrease access latency within a system would have modified Balakrishnan in view of Hughes to cluster cores—and particularly couple cores and cache portions—based on physical proximity. Ex-1002 ¶¶197–98.

4. Claim 12/Claim 13/Claim 14

Balakrishnan discloses that “[t]he banks of NUCA cache 135 may be arranged according to a distance hierarchy with respect to [its] core[s]” and that “[t]he distance hierarchy may refer to the several levels of delay or access time.” Ex-1006 ¶31. Further, Balakrishnan discloses that “caches may contain memory banks that are connected with wires, traces, or interconnections,” which “introduce

various delays.” *Id.* 41; *see id.* ¶31.

Moreover, Balakrishnan discloses that “[b]anks may generally be assigned to different groups based on access latencies.” Ex-1006 ¶10. Accordingly, a POSITA would have understood that clustering adjoining or physically proximate cores would decrease access latency within the cluster and allow for frequency control or power gating at the cluster level. Ex-1002 ¶¶199–202. Thus, a POSITA would have been motivated to implement a first interconnect with first characteristics (e.g., lower access latency and higher data throughput) to couple the cores of a proximate cluster and their local cache portions while maintaining access to each cache portion by all of the non-proximate clusters over a second interconnect with second characteristics (e.g., different access latency and/or data throughput). *Id.*

To the extent that Balakrishnan does not explicitly teach a first and second interconnect, Balakrishnan in view of Hughes renders this limitation obvious. *Id.* Hughes’s Figure 5 (annotated) depicts “a non-uniform cache architecture,” in which “a number of cache molecules 510–518 [(yellow)] and processor cores 520–528 [(blue)] may be interconnected with a dual ring interconnect [(i.e., an on-die interconnect) (purple)], having a clockwise ring 552 and a counter-clockwise ring 550.” Ex-1008 ¶34; Ex-1002 ¶¶199–202.

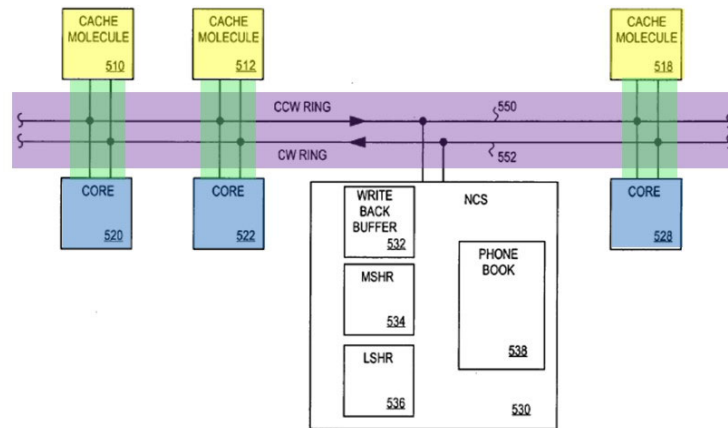


FIG. 5

Ex-1008, Fig. 5.

In addition, Figure 5 illustrates how each core is paired with a cache molecule (i.e., a cache portion) over a local core-cache interconnect (green). Ex-1008 ¶18. “The pairing is to logically associate a core with the ‘closest’ cache molecule in terms of low access latency.” *Id.* ¶18, code (57) (“Each of the cache molecules may be closest, in terms of access latency time, to one of the processor cores.”). Accordingly, a POSITA would have understood Hughes to teach a first and second interconnect with first and second characteristics (e.g., access latency and/or data throughput). Ex-1002 ¶¶199–202. Therefore, a POSITA would have been motivated to further reduce access latencies of Balakrishnan by using separate interconnects to couple cache portions with proximate and non-proximate clusters of cores. *Id.*

VIII. THE BOARD SHOULD NOT USE ITS DISCRETION TO DENY INSTITUTION UNDER *FINTIV*

The Board should decline to exercise its discretion to deny institution under 35 U.S.C. § 314(a). While Director Vidal’s June 21, 2022, Guidance made clear that “*Fintiv* is limited to facts of that case,” for completeness, Petitioner addresses the factors set forth in *Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 (PTAB Mar. 20, 2020) (precedential). The six factors set out in *Fintiv* do not justify denying institution.

The **first *Fintiv* factor** is neutral because Samsung has not yet moved to stay the district court proceedings pending IPR, and the PTAB will not infer how the district court would rule should a stay be requested. *See, e.g., Hulu, LLC v. SITO Mobile R&D IP, LLC*, IPR2021-00298, Paper 11 at 10–11 (PTAB May 19, 2021).

For the **second *Fintiv* factor**, the district court has set a trial date in the Daedalus matter for 5/20/2024. This timing does not tip in favor of denying the petition in this case. *See, e.g., Apple Inc. v. Koss Corp.*, IPR2021-00305, Paper 14 at 13 (PTAB June 3, 2021). This Petition is filed in March 2023. An institution decision should be issued within six months of docketing, around September 2023. And a final written decision (“FWD”) should be issued in September 2024. Thus, the trial date is likely less than four months before the FWD. Recent PTAB cases analyzing similar schedules have weighed the second factor as neutral or, at most, slightly in favor of denial. *See, e.g., Google LLC et al. v. Parus Holdings, Inc.*,

IPR2020-00847, Paper 9 at 12 (PTAB Oct. 21, 2020) (trial scheduled for three months before FWD weighs “at best, neutral”); *Apple Inc. v. Koss Corp.*, IPR2021-00305, Paper 14 at 13-14 (PTAB June 3, 2021) (trial scheduled for two months before FWD weighs “at most, minimally in favor” of denial).

“The PTAB will also consider additional supporting factors such as the number of cases before the judge in the parallel litigation.” Director Vidal’s June 2022 Guidance. Here, the court has scheduled three other trials (2:22-cv-00312-JRG-RSP, *Fleet Connect Solutions LLC v. Southern Tire Mart, LLC*; 2:22-cv-00322-JRG-RSP, *Lionra Technologies Limited v. Fortinet, Inc.*; and 2:22-cv-00361-JRG-RSP S3G, *Technology LLC v. Nordstrom Inc.*) in the same week as the trial for this case.

If the median time to trial for civil actions in the Eastern District of Texas (“EDTX”) is used to estimate the trial date (i.e., to account for potential delays, etc.), per Director Vidal’s June 2022 Guidance, trial should not be expected to occur until October 2024, at least a month after a FWD. Ex-1017 at 35 (indicating a most-recent median time to trial for civil actions in EDTX of 24.5 months).

The **third *Fintiv* factor** weighs strongly against denial. In particular, Samsung has not yet answered Daedalus’s Complaint, and has instead moved to dismiss. No hearing has been set and no substantive activity has occurred in the case. Because the investment in the trial has been minimal and Petitioner acted

diligently, this factor weighs strongly against discretionary denial. *See, e.g., Hulu*, Paper 11 at 13.

The **fourth *Fintiv* factor** also weighs against denial. As explained above, the median time to trial for civil actions in EDTX suggests that trial will not occur until at least a month after a FWD. Thus, “there is a reasonable likelihood that the Board will address the overlapping validity issues prior to the district court reaching them at trial..., thereby providing the possibility of simplifying issues for trial.” *Juniper Networks, Inc. v. Packet Intelligence LLC*, IPR2020-0039, Paper 21 at 18 (PTAB Sept. 10, 2020); *see also MED-EL Elektromedizinische Geräte GmbH v. Sonova AG*, IPR2020-00176, Paper 13 at 15 (PTAB June 3, 2020) (“As to the fourth factor,...overlap exists between the invalidity issues in this case and in the district court. This overlap may inure to the district court’s benefit, however, by simplifying issues for trial should we reach our determination on the challenges raised in the Petition before trial.”).

Moreover, Petitioner is challenging more claims than asserted in the district court litigation. Patent Owner’s infringement contentions assert only Claims 1–3, 11, 15–21. This Petition challenges Claims 1–4, 6, 11–17, 20, 21. Thus, this IPR is the only venue in which the invalidity challenges against several claims in the ’960 Patent will be adjudicated. Moreover, reviewing the challenged claims in this proceeding promotes efficiency because many more claims will be addressed here,

which will resolve disputes against future district court defendants. *See Fintiv*, Paper 11 at 4 (“[T]he Board’s cases addressing earlier trial dates as a basis for denial under NHK have sought to balance considerations such as system efficiency, fairness, and patent quality.”). Accordingly, challenging additional claims further weighs against denial. *See, e.g., Vudu, Inc. v. Ideahub, Inc.*, IPR2020-01688, Paper 16 at 14–15 (PTAB Apr. 19, 2021) (“differences in asserted claims” and claims challenged in the petition weighs against denial).

On the **fifth *Fintiv* factor**, the Board should give little weight to the fact that Petitioner and PO are the same parties as in district court. Although Samsung is both petitioner and litigation defendant, this factor does not outweigh the other factors that strongly weigh against discretionary denial. *See Weatherford U.S., L.P., v. Enventure Global Tech., Inc.*, IPR2020-01684, Paper 16 at 11–13 (PTAB Apr. 14, 2021).

“Other circumstances” under the **sixth *Fintiv* factor** further weigh against discretionary denial. The Petition’s merits are strong with two primary references, Moses and Balakrishnan, alone or in combination with one or two secondary references, rendering independent Claims 1, 15, 20 and dependent Claims 2–14, 16–19, 21 obvious. *See Align Technology, Inc. v. 3Shape A/S*, IPR2020-01087, Paper 15 at 42–43 (PTAB Jan. 20, 2021); *see also supra* Section VII.

Taken together holistically, the *Fintiv* factors strongly weigh against

exercising discretion to deny institution.

IX. CONCLUSION

The unpatentability grounds presented above are reasonably likely to prevail, and IPR should be instituted for Claims 1–4, 6, 11–17, 20, 21 of the '960 Patent based on each of the grounds specified in this Petition.

Respectfully Submitted,

/s/ John Kappos

John Kappos (Reg. No. 37,861)

CERTIFICATE OF WORD COUNT

Under 37 C.F.R. § 42.24(d), Petitioner certifies that this petition includes 13,985 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 March 8, 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:

Vecchia/NDWE/Intel
99 Almaden Boulevard, Suite 575
San Jose, CA 95113

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

Garland T. Stephens
Blue Peak Law Group
2007 South Boulevard
Houston, TX 77098
Phone: (832) 387-7696
Email: garland@bluepeak.law

Anna Dwyer
Blue Peak Law Group
345 East 94th Street, Ste Apt 28g
New York, NY 10128
Phone: (772) 631-6460
Email: anna@bluepeak.law

Jeff Risher
Blue Peak Law Group
15226 Bluff View Ave.
Friant, CA 93626
Phone: (949) 690-1462
Email: jeff@bluepeak.law

Justin Lile Constant
Blue Peak Law Group

3139 W Holcombe Blvd., PMB 8160
Houston, TX 77025
Phone: (281) 972-3036
Email: justin@bluepeak.law

Kate S. Falkenstien
Robert Stephen Magee
Blue Peak Law Group
3790 El Camino Real, Ste PMB 846
Palo Alto, CA 94306
Phone: (281) 972-3036
Email: kate@bluepeak.law
Email: robert@bluepeak.law

Richard M. Koehl
Blue Peak Law Group
PO Box 20204
New York, NY 10001-0006
Phone: (917) 856-3872
Email: richard@bluepeak.law

Andrea Leigh Fair
Claire Abernathy Henry
Charles Everingham , IV
Ward, Smith & Hill, PLLC
1507 Bill Owens Parkway
Longview, TX 75604
Phone: (903) 757-2323
Email: andrea@wsfirm.com
Email: claire@wsfirm.com
Email: ce@wsfirm.com

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

Dated: March 8, 2023

Respectfully submitted,

/s/ John Kappos

John Kappos (Reg. No. 37,861)
O'Melveny & Myers LLP
2501 North Harwood Street, Suite 1700
Dallas, TX 75201
Telephone: (972) 360-1900
Fax: (972) 360-1901
E-Mail: jkappos@omm.com
Attorney for Petitioner