

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

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

v.

DAEDALUS PRIME LLC
Patent Owner.

**DECLARATION OF DR. STEPHEN W. MELVIN
IN SUPPORT OF PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 10,372,197**

TABLE OF CONTENTS

	Page
I. INTRODUCTION, BACKGROUND, QUALIFICATIONS, AND INFORMATION CONSIDERED.....	1
A. INTRODUCTION.....	1
B. BACKGROUND AND QUALIFICATIONS	2
C. INFORMATION CONSIDERED	4
II. RELEVANT LEGAL STANDARDS.....	5
A. Claim Interpretation	5
B. Written Description.....	5
C. Perspective of One of Ordinary Skill in the Art.....	6
D. Obviousness.....	6
III. RELIEF REQUESTED AND CLAIM CONSTRUCTION.....	9
IV. THE CHALLENGED PATENT	10
A. The '197 Patent	10
B. Prosecution History of the '197 Patent	12
C. Level of Ordinary Skill in the Art.....	13
V. BRIEF DESCRIPTION OF THE APPLIED PRIOR ART REFERENCES	13
A. Naveh (Ex-1005).....	13
B. Saeed (Ex-1006).....	15
C. Titiano (Ex-1007).....	16
D. Hansquine (Ex-1008)	19
VI. TECHNOLOGICAL BACKGROUND	21
A. It was well-known to use an interconnect to couple cores in a multi-core processor to a shared cache memory.....	21
B. It was well-known to monitor the workload of cores in a multi-core processor.....	23

TABLE OF CONTENTS
(continued)

	Page
C. It was well-known to control the frequency and voltage of cores to optimize for power or performance and use a table to store voltage and frequency values.	26
VII. DETAILED EXPLANATION OF THE UNPATENTABILITY GROUNDS	29
A. Grounds 1 and 2: Claims 1–3 and 19–20 Are Obvious Over Naveh and Obvious Over Naveh in view of Saeed.....	29
1. A POSITA would have been motivated to combine Naveh with Saeed.....	29
2. Independent Claim 1	32
a. Element 1[preamble]: A processor comprising:.....	32
b. Element 1[a]: a plurality of cores;	33
c. Element 1[b]: a cache memory;.....	35
d. Element 1[c]: an interconnect to couple the plurality of cores to the cache memory;	35
e. Element 1[d]: a power controller to control a plurality of power management features of the processor;	36
f. Element 1[e]: wherein the power controller includes a tuning circuit to [e1] receive a workload configuration input regarding a workload, [e2] receive a plurality of energy performance bias (EPB) values and determine a global EPB value based thereon, and [e3] update at least one setting of at least one of the plurality of power management features based on the workload configuration input and the global EPB value.....	38
i. Element 1[e1]: a tuning circuit to receive a workload configuration input regarding a workload;.....	39

TABLE OF CONTENTS

(continued)

	Page
ii. Element 1[e2]: a tuning circuit to ... receive a plurality of energy performance bias (EPB) values and determine a global EPB value based thereon; 44 iii. Element 1[e3]: a tuning circuit to ... update at least one setting of at least one of the plurality of power management features based on the workload configuration input and the global EPB value. 51 3. Dependent Claims 2–3, 19–20.....52 a. Claim 2: The processor of Claim 1, wherein the power controller is to access a power-performance tuning table based on the workload configuration input and the global EPB value, and use information from the power-performance tuning table to update the at least one setting. 52 b. Claim 3: The processor of claim 2, wherein the tuning circuit is to update the at least one setting of the at least one power management feature responsive to the global EPB value. 57 c. Claim 19: The processor of claim 1, wherein the power controller is to enable a first power management feature based on the updated at least one setting. 57 d. Claim 20: The processor of claim 1, wherein the power controller is to send an update message to a destination agent of the processor to cause the destination agent to control a first power management feature based on the updated at least one setting. 58 B. Ground 3: Claims 1–6, 11, 13, and 19–20 Are Obvious Over Titiano.....60 1. Independent Claim 1 60	

TABLE OF CONTENTS

(continued)

	Page
a. Element 1[preamble]: A processor comprising:.....	60
b. Element 1[a]: a plurality of cores;	61
c. Element 1[b]: a cache memory;.....	65
d. Element 1[c]: an interconnect to couple the plurality of cores and the cache memory;.....	66
e. Element 1[d]: a power controller to control a plurality of power management features of the processor;	67
f. Element 1[e]: wherein the power controller includes a tuning circuit to [e1] receive a workload configuration input regarding a workload, [e2] receive a plurality of energy performance bias (EPB) values and determine a global EPB value based thereon, and [e3] update at least one setting of at least one of the plurality of power management features based on the workload configuration input and the global EPB value.....	70
i. Element 1[e1]: a tuning circuit to receive a workload configuration input regarding a workload;.....	71
ii. Element 1[e2]: a tuning circuit to ... receive a plurality of energy performance bias (EPB) values and determine a global EPB value based thereon;	77
iii. Element 1[e3]: a tuning circuit to ... update at least one setting of at least one of the plurality of power management features based on the workload configuration input and the global EPB value.	87
2. Dependent Claims 2–6 and 19–20	94

TABLE OF CONTENTS
(continued)

	Page
a. Claim 2: The processor of claim 1, wherein the power controller is to access a power-performance tuning table based on the workload configuration input and the global EPB value, and use information from the power-performance tuning table to update the at least one setting.	94
b. Claim 3: The processor of claim 2, wherein the tuning circuit is to update the at least one setting of the at least one power management feature responsive to the global EPB value.	98
c. Claim 4: The processor of claim 3, wherein the tuning circuit includes a sampler to receive the plurality of EPB values from at least some of the plurality of cores and a combiner to generate the global EPB value from the EPB value from the plurality of cores.	98
d. Claim 5: The processor of claim 4, wherein the sampler is to receive the EPB value from a plurality of threads to execute on the at least some of the plurality of cores.	99
e. Claim 6: The processor of claim 4, wherein the power controller is to generate a bin value from the global EPB value and to access the power-performance tuning table using the bin value.	100
f. Claim 19: The processor of claim 1, wherein the power controller is to enable a first power management feature based on the updated at least one setting.	101
g. Claim 20: The processor of claim 1, wherein the power controller is to send an update message to a destination agent of the processor to cause the destination agent to control a first power management feature based on the updated at least one setting.	101

TABLE OF CONTENTS
(continued)

	Page
3. Independent Claim 11	105
a. Element 11[pre]: A non-transitory machine-readable medium having stored thereon instructions, which if performed by a machine cause the machine to perform a method comprising:	105
b. Element 11[a]: receiving an energy performance bias (EPB) value in a tuning circuit of a power control logic of a processor from a plurality of threads executing on the processor and determining a global EPB value based at least in part on the EPB value received from the plurality of threads, the EPB value to indicate a preference of a user for a tradeoff between power optimization and performance optimization;.....	106
c. Elements 11[b]: accessing a table based on the global EPB value; and Element 11[c]: updating at least one setting of a power management feature controlled by the power control logic based on information obtained from the table.	107
4. Dependent Claim 13: The non-transitory machine-readable medium of claim 11, wherein the method further comprises providing an interface to enable a user, an operating system or a baseboard management controller to set the EPB value, and controlling a plurality of power management features via input of the EPB value.	107
C. Ground 4: Claims 2 and 15 Are Obvious Over Titiano in View of Hansquine.....	108
1. A POSITA would have been motivated to combine Titiano and Hansquine	108
2. Dependent Claim 2.....	109
3. Independent Claim 15	111

TABLE OF CONTENTS
(continued)

	Page
a. Element 15[a]: a multicore processor including a plurality of cores, at least one cache memory and a memory controller;	111
b. Element 15[b]: wherein the tuning circuit is to access an entry of a tuning table based at least in part on the EPB value and a workload configuration value and update a setting for a power management feature responsive to a value stored in a field of the entry;.....	112
c. Element 15[c]: wherein the workload configuration value is to indicate a predominant workload type to be executed on the system, the tuning table including a first set of entries associated with a first workload type and a second set of entries associated with a second workload type;	113
d. Element 15[d]: a system memory coupled to the multicore processor via a memory interconnect.....	114
4. Dependent Claim 16: The system of claim 15, wherein the tuning circuit is to update at least one setting of a first power management feature controlled by a power control logic of the multicore processor, the at least one setting obtained from a field of an entry of the tuning table associated with a bin in which the EPB value is included.....	115
5. Dependent Claim 17: The system of claim 16, wherein multicore processor comprises the tuning circuit.	116
6. Dependent Claim 18: The system of claim 15, wherein the multicore processor comprises a power control logic, the power control logic comprising the tuning circuit, wherein the tuning circuit is to generate a bin value from the EPB value and access the tuning table using the bin value.	116
VIII. CONCLUSION.....	117

LIST OF EXHIBITS¹

Ex-1001	U.S. Patent No. 10,372,197 (“the ’197 Patent”)
Ex-1002	Declaration of Dr. Stephen W. Melvin
Ex-1003	Curriculum Vitae of Dr. Stephen W. Melvin
Ex-1004	Prosecution History of U.S. Patent No. 10,372,197 (App. No. 15/367,330)
Ex-1005	U.S. Patent Application Publication No. 2006/0026447 (“Naveh”)
Ex-1006	U.S. Patent No. 6,711,447 (“Saeed”)
Ex-1007	U.S. Patent Application Publication No. 2008/0162770 (“Titiano”)
Ex-1008	U.S. Patent Application Publication No. 2011/0004774 (“Hansquine”)
Ex-1009	Bruce Jacob et al., <i>Memory Systems, Cache, DRAM, Disk</i> (2008) (“Jacob”)
Ex-1010	Ahmed Amine Jerraya and Wayne Wolf, <i>Multiprocessor Systems-on-Chips</i> (2005) (“Jerraya”)
Ex-1011	Brinkley Sprunt, <i>The Basics Of Performance-monitoring Hardware; Pentium 4 Performance-Monitoring Features</i> (2002) (“Sprunt”)
Ex-1012	Dmitrijs Zaparanuks et al., <i>Accuracy of Performance Counter Measurements</i> (2009) (“Zaparanuks”)
Ex-1013	Paul West, <i>Core Monitors: Monitoring Performance In Multicore Processors</i> (2008) (“West”)

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.

Ex-1014	K. Sreejith, <i>Workload Based Lookup Table for Minimal Power Operation Under Supply and Body Bias Control</i> (2009) (“Sreejith”)
Ex-1015	<i>Enhanced Intel® SpeedStep® Technology for the Intel® Pentium® M Processor</i> (2004) (“Intel SpeedStep”)
Ex-1016	INTENTIONALLY LEFT BLANK
Ex-1017	U.S. District Courts – Combined Civil and Criminal Federal Court Management Statistics (June 30, 2022), available at: https://www.uscourts.gov/statistics-reports/federal-court-management-statistics-june-2022
Ex-1018	INTENTIONALLY LEFT BLANK
Ex-1019	INTENTIONALLY LEFT BLANK

I. INTRODUCTION, BACKGROUND, QUALIFICATIONS, AND INFORMATION CONSIDERED

I, Stephen W. Melvin, declare as follows:

A. INTRODUCTION

1. I have been retained by Samsung Electronics Co., Ltd. and Samsung Electronics America, Inc. (collectively, “Petitioner”) as an independent expert consultant in *inter partes* review (“IPR”) proceeding before the United States Patent and Trademark Office (“USPTO”).

2. I have been asked by Samsung Counsel (“Counsel”) to consider whether certain references teach or suggest the features recited in Claims 1–6, 11, 13, and 15–20 of U.S. Patent No. 10,372,197 (“the ’197 Patent”) (Ex-1001), which I understand is allegedly owned by Daedalus Prime LLC (“Patent Owner”). My opinions and the bases for my opinions are set forth below.

3. My employer is being compensated for services provided in this matter at my customary rate of \$650 per hour, plus travel expenses. My compensation is not conditioned on the conclusions I reach as a result of my analysis or on the outcome of this matter. Similarly, my compensation is not dependent upon and in no way affects the substance of my statements in this declaration.

4. I have no financial interest in Petitioner or any of its subsidiaries. I do

not have any financial interest in the '197 Patent and have not had any contact with any of the named inventors of the '197 Patent.

B. BACKGROUND AND QUALIFICATIONS

5. I received a Ph.D. in Computer Science from the University of California at Berkeley in 1991 and a B.S. in Electrical Engineering and Computer Science from the University of California at Berkeley in 1982. I have more than 40 years of experience in computer science and computer engineering. I am an inventor on over 45 patents, and I am a registered patent agent before the USPTO.

6. My Ph.D. research areas included high-performance computer architecture and microarchitecture and microcode-based system performance analysis tools. From September 2001 through April 2002, I was a Visiting Scholar at the University of Texas at Austin, where I directed graduate students in research in the area of high-performance computer architecture.

7. In May 2001, I co-founded and was the Chief Architect of Flowstorm, Inc., a start-up company based in Silicon Valley, where I defined and guided the overall chip architecture for a multithreaded packet processor. From March 2000 through May 2001, I worked as the Senior CPU Architect at Clearwater Networks, where I was involved in defining the architecture and microarchitecture of Clearwater's CNP810S multithreaded network processor. Both of those professional experiences required a detailed understanding of processor design

including chips having multiple processor cores and power optimization features.

8. From 1983 to 2020, I was the President of Zytek Communications Corporation (“Zytek”). Zytek was an engineering, consulting, and small-scale manufacturing company that provided intellectual property consulting services as well as services related to the design, implementation, and testing of embedded systems. Zytek’s general areas of activity included industrial control and measurement, Internet-related services, hard disk analysis and file recovery, and computer engineering research services. Through my work at Zytek, I have designed numerous microprocessor-based embedded systems, including analog and digital circuit design, firmware development for embedded microcontrollers, and software development for host interfacing, product development, and debugging. My designs for embedded systems have all involved an understanding of designing with microprocessors and incorporating microprocessors into complete systems.

9. I am currently employed as a Principal with Exponent, Inc., which is an international multidisciplinary engineering and scientific consulting firm.

10. I am a member of the Institute of Electrical and Electronics Engineers (IEEE) and the Association for Computing Machinery (ACM).

11. I served as General Chair of the 45th Annual International Symposium on Microarchitecture (Micro-45), held in Vancouver in December of

2012. I also served as co-chair of the 29th Annual International Symposium on Microarchitecture (Micro-29), held in Paris in December of 1996.

12. I have authored numerous peer reviewed publications relating to processor design. I have also supervised graduate students performing work relating to processor design. I am also the named inventor on numerous patents and patent applications, a number of which relate to the design and fabrication of computing devices. My curriculum vitae, which is Exhibit 1003, contains a complete list of my publications and patents.

13. Additional details about my employment history, fields of expertise, awards, publications, and other activities are further included in my CV, attached as Exhibit 1003.

C. INFORMATION CONSIDERED

14. In preparation for this declaration, I have considered the materials discussed in this declaration, including, for example, the '197 Patent, the references cited by the '197 Patent, the prosecution history of the '197 Patent and its parent applications (including the references cited therein), various background articles and materials referenced in this declaration, and the prior art references identified in this declaration. In addition, my opinions are further based on my education, training, experience, and knowledge in the relevant field.

II. RELEVANT LEGAL STANDARDS

15. I am not an attorney and offer no legal opinions. For the purposes of this Declaration, I have been informed about certain aspects of the law that are relevant to my analysis, as summarized below.

A. Claim Interpretation

16. I have been informed and understand that in an IPR proceeding, claims are to be interpreted according to the *Phillips* claim construction standard. I have been informed and understand that claim construction is a matter of law and that the final claim constructions for this proceeding will be determined by the Patent Trial and Appeal Board (“PTAB”).

B. Written Description

17. I understand that determining whether a particular patent or printed publication constitutes prior art to a challenged patent claim can require determining the effective filing date (also known as the priority date) to which the challenged claim is entitled. I understand that for a patent claim to be entitled to the benefit of the filing date of an earlier application to which the patent claims priority, the earlier application must have described the claimed invention in sufficient detail to convey with reasonable clarity to the POSITA that the inventor had possession of the claimed invention as of the earlier application’s filing date. I understand that a disclosure that merely renders the claimed invention obvious is

not sufficient written description for the claim to be entitled to the benefit of the filing date of the application containing that disclosure.

C. Perspective of One of Ordinary Skill in the Art

18. I have been informed and understand that a patent is to be understood from the perspective of a hypothetical “person of ordinary skill in the art” (“POSITA”). Such an individual is considered to possess normal skills and knowledge in a particular technical field (as opposed to being a genius). I understand that in considering what the claims of a patent require, what was known prior to that patent, what a prior art reference discloses, and whether an invention is obvious or not, one must use the perspective of such a POSITA.

D. Obviousness

19. I have been informed and understand that a patent claim is obvious under 35 U.S.C. § 103, and therefore invalid, if the claimed subject matter, as a whole, would have been obvious to a POSITA as of the priority date of the patent based on one or more prior art references and/or the knowledge of a POSITA.

20. I understand that an obviousness analysis must consider (1) the scope and content of the prior art, (2) the differences between the claims and the prior art, (3) the level of ordinary skill in the pertinent art, and (4) secondary considerations, if any, of non-obviousness (such as unexpected results, commercial success,

long-felt but unmet need, failure of others, copying by others, and skepticism of experts).

21. I understand that a prior art reference may be combined with other references to disclose each element of the invention under 35 U.S.C. § 103. I understand that a reference may also be combined with the knowledge of a POSITA, and that this knowledge may be used to combine multiple references. I further understand that a POSITA is presumed to know the relevant prior art. I understand that the obviousness analysis may take into account the inferences and creative steps that a POSITA would employ.

22. In determining whether a prior art reference would have been combined with other prior art or other information known to a POSITA, I understand that the following principles may be considered:

- whether the references to be combined involve non-analogous art;
- whether the references to be combined are in different fields of endeavor than the alleged invention in the Patent;
- whether the references to be combined are reasonably pertinent to the problems to which the inventions of the Patent are directed;
- whether the combination is of familiar elements according to known methods that yields predictable results;

- whether a combination involves the substitution of one known element for another that yields predictable results;
- whether the combination involves the use of a known technique to improve similar items or methods in the same way that yields predictable results;
- whether the combination involves the application of a known technique to a prior art reference that is ready for improvement, to yield predictable results;
- whether the combination is “obvious to try”;
- whether the combination involves the known work in one field of endeavor prompting variations of it for use in either the same field or a different one based on design incentives or other market forces, where the variations are predictable to a POSITA;
- whether there is some teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill in the art to modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention;
- whether the combination requires modifications that render the prior art unsatisfactory for its intended use;
- whether the combination requires modifications that change the principle of operation of the reference;

- whether the combination is reasonably expected to be a success; and
- whether the combination possesses the requisite degree of predictability at the time the invention was made.

23. I understand that in determining whether a combination of prior art references renders a claim obvious, it is helpful to consider whether there is some teaching, suggestion, or motivation to combine the references and a reasonable expectation of success in doing so. I understand, however, that a teaching, suggestion, or motivation to combine is not required.

III. RELIEF REQUESTED AND CLAIM CONSTRUCTION

24. For purposes of this declaration, I have assumed that the '197 Patent is entitled to its earliest alleged domestic priority date, December 15, 2011, and thus apply pre-AIA 35 U.S.C. § 102.

25. It is my opinion that Claims 1–6, 11, 13, and 15–20 of the '197 Patent would have been obvious under 35 U.S.C. § 103 on the following grounds.

Ground	Summary
1	Claims 1–3 and 19–20 are obvious over Naveh (Ex-1005).
2	Claims 1–2 are obvious over Naveh in view of Saeed (Ex-1006).
3	Claims 1–6, 11, 13, and 19–20 are obvious over Titiano (Ex-1007).
4	Claims 2 and 15–18 are obvious over Titiano in view of Hansquine (Ex-1008).

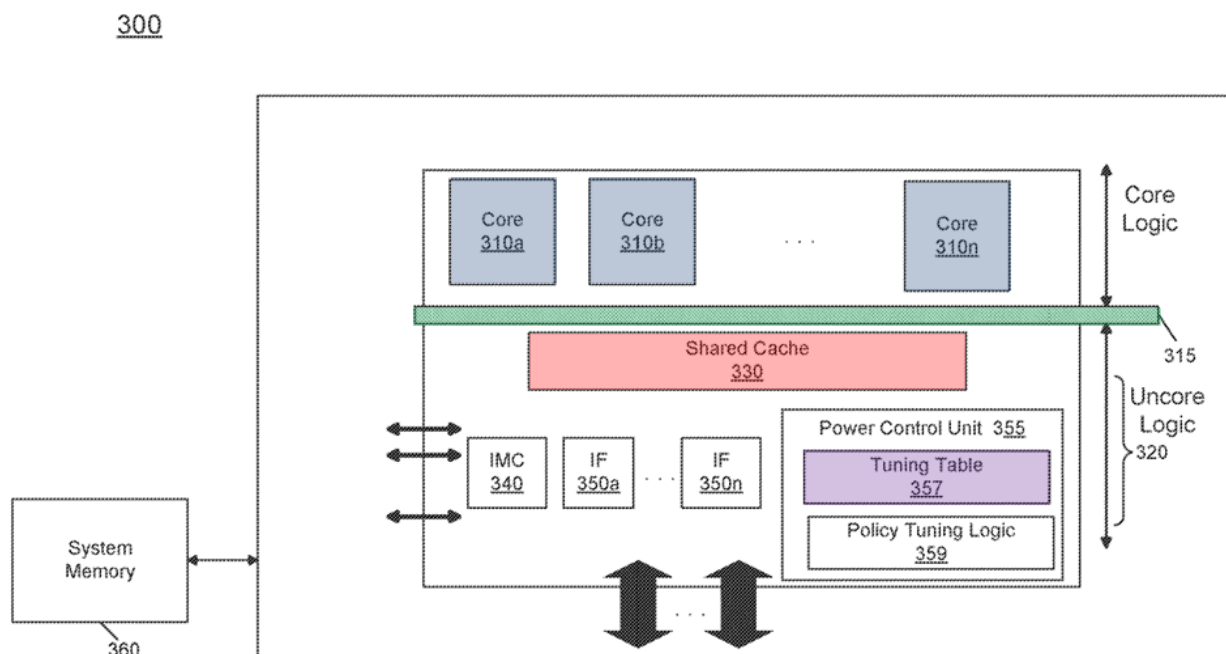
26. I do not believe that any term requires explicit construction to resolve the issues presented in this Declaration. I ascribe the plain meaning to each claim term, as that plain meaning would have been understood by a POSITA.

27. I reserve the right to offer opinions on any claim constructions proposed in this proceeding or to offer opinions on additional constructions in the district court.

IV. THE CHALLENGED PATENT

A. The '197 Patent

28. The alleged invention of the '197 Patent relates to the control of power management features. With reference to annotated Figure 4 below, the claims recite a processor 300; a plurality of cores 310; a cache memory 330; and an interconnect 315 to couple the plurality of cores and the cache memory. Ex-1001, 8:66–9:10. The '197 Patent does not suggest that this arrangement of components is inventive.



29. Rather, the '197 purports to describe an allegedly inventive tuning circuit for controlling power management features. Specifically, processor 300 may include a tuning circuit to (1) receive a workload configuration input and a plurality of energy performance bias (EPB) values, (2) determine a global EPB value based on the plurality of EPB values, and (3) update at least one setting of at least one of a plurality of power management features based on the workload configuration input and the global EPB value. *Id.*, 11:49–62.

30. The system may also include a power-performance tuning table that may be used to update at least one setting of at least one of a plurality of power management features. *Id.*, 5:33–7:31. The '197 Patent also describes using the

workload configuration input and the global EPB value to access a power-performance tuning table. *Id.*, 3:2–51. The power-performance tuning table may contain information used to update at least one setting of at least one of a plurality of power management features. *Id.*, 6:12–30.

B. Prosecution History of the '197 Patent

31. The '197 file history is submitted as Ex-1004. The '197 Patent was filed on December 2, 2016, as a continuation to Application No. 14/855,553 (September 16, 2015), which is a continuation of Application No. 13/782,473 (March 1, 2013), which is a continuation of Application No. 13,326,586 (December 15, 2011). The '197 Patent alleges priority to Application No. 13,326,586, filed on December 15, 2011. Ex-1001, code (63).

32. The Examiner issued a first non-final rejection, mailed December 29, 2017, rejecting the claims on three grounds: (1) non-statutory double patenting; (2) as being directed to ineligible subject matter; and (3) as being obvious Ex-1004, 210–16. The Applicant amended its claims on April 25, 2018, and argued that the claims were definite. *Id.*, 185–190. The Applicant filed a terminal disclaimer to overcome the double patenting rejection on April 25, 2018. *Id.*, 193–96.

33. The Examiner issued a second non-final rejection on August 15, 2018, again rejecting the claims as being directed to ineligible subject matter. *Id.*, 175–79. The Applicant amended its claims again on November 2, 2018, and argued that

the claims were not directed to ineligible subject matter. *Id.*, 163–172.

34. A notice of allowance issued March 27, 2019, but no substantive explanation was offered by the Examiner. *Id.*, 29–31. Nonetheless, the claimed subject matter was well-known in the art, as shown below in Section VII.

C. Level of Ordinary Skill in the Art

35. 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 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.

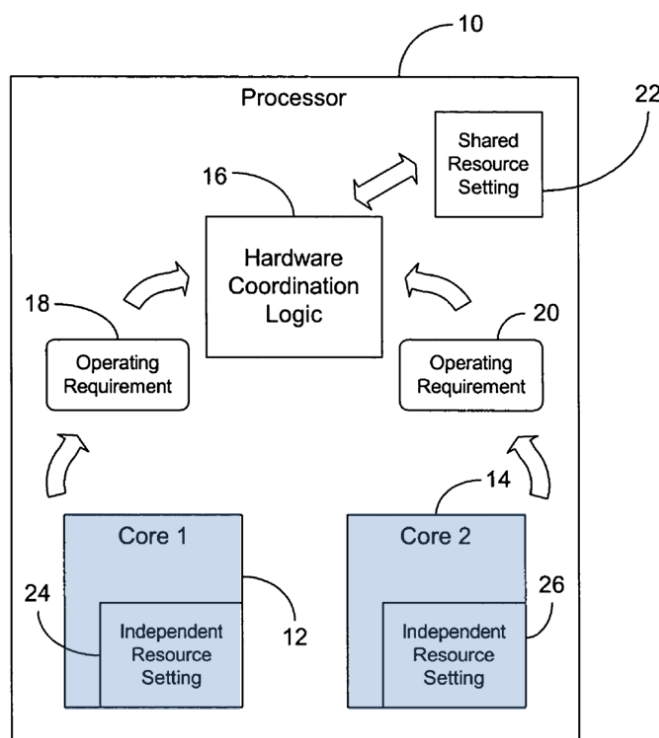
V. BRIEF DESCRIPTION OF THE APPLIED PRIOR ART REFERENCES

A. Naveh (Ex-1005)

36. Naveh was filed July 27, 2004, and published February 2, 2006. Ex-1005, code (43). Naveh qualifies as prior art under at least pre-AIA 35 U.S.C. § 102(a), (b), and (e).

37. Naveh describes “systems and methods of managing power [in a multi-core processor] ... [by] issuing a first operating requirement from a first processor core and issuing a second operating requirement from a second processor core.” Ex-1005, code (57). As shown in Figure 1, Naveh’s “processor 10 ha[s] a

first core 12, a second core 14 and hardware coordination logic 16 operatively coupled to the first core 12 and the second core 14.” *Id.* ¶17.



Further, “the illustrated processor 10 may have a processor clock frequency setting and a power supply voltage setting that are shared by both cores 12, 14,” therefore “the clock frequency and power supply voltage settings could be viewed as a shared resource setting 22.” *Id.* ¶20.

38. In Naveh, hardware coordination logic 16 receives and compares operating requirements 18 and 20 and selects one of them to be shared resource setting 22. *Id.* ¶¶26–27. Which operating requirement the coordination logic selects depends, in part, on the operating mode of the system. *Id.* ¶¶26, 39. Naveh refers to

these modes as a “max detect” mode (i.e., a performance/maximum policy) and a “minimum-of-both-requests” mode (i.e., a power/minimum policy). *Id.* ¶¶23, 27, 33. Thus, coordination logic 16 selects the lower or the higher of the two operating requirements based on whether “performance or a power policy” is in place. *Id.* ¶¶23, 27.

B. Saeed (Ex-1006)

39. Saeed was filed January 22, 2003, and published March 23, 2004. Saeed qualifies as prior art under pre-AIA 35 U.S.C. § 102(a), (b), and (e).

40. Saeed is directed to “control[ling] the frequency and voltage of a central processing unit (CPU) of a multi-core CPU architecture environment based on the measured levels of multi-threadedness.” Ex-1006, code (57). Saeed teaches that workload may be determined by the level of multi-threadedness. *Id.*, 2:9–11 (“For one embodiment of the invention, the workload may be defined by the level of multi-threadedness.”), *id.*, 3:13–16 (“The power control circuit 210 receives as input a measured workload from the operating system. The workload may be defined by the level of multi-threadedness of the IC 200.”). Power control circuit 210 may also select different power/performance modes based on the workload configuration input. *Id.*, 4:24–35.

41. Saeed further teaches that power control circuit 210 may use the workload configuration input (e.g., the level of multi-threadedness) to index a

“lookup table” to identify a “corresponding” frequency or voltage setting for the cores 220, 230, and 240:

The frequency setting may be a function of the level of multi-threadedness measured in operation 410. Depending on the level of multi-threadedness of the system, *a lookup table* or a logic circuit may be used to determine a corresponding frequency that is between the minimum frequency and the maximum frequency. Similarly, voltages of the CPU cores may be adjusted based on the level of multi-threadedness measured.

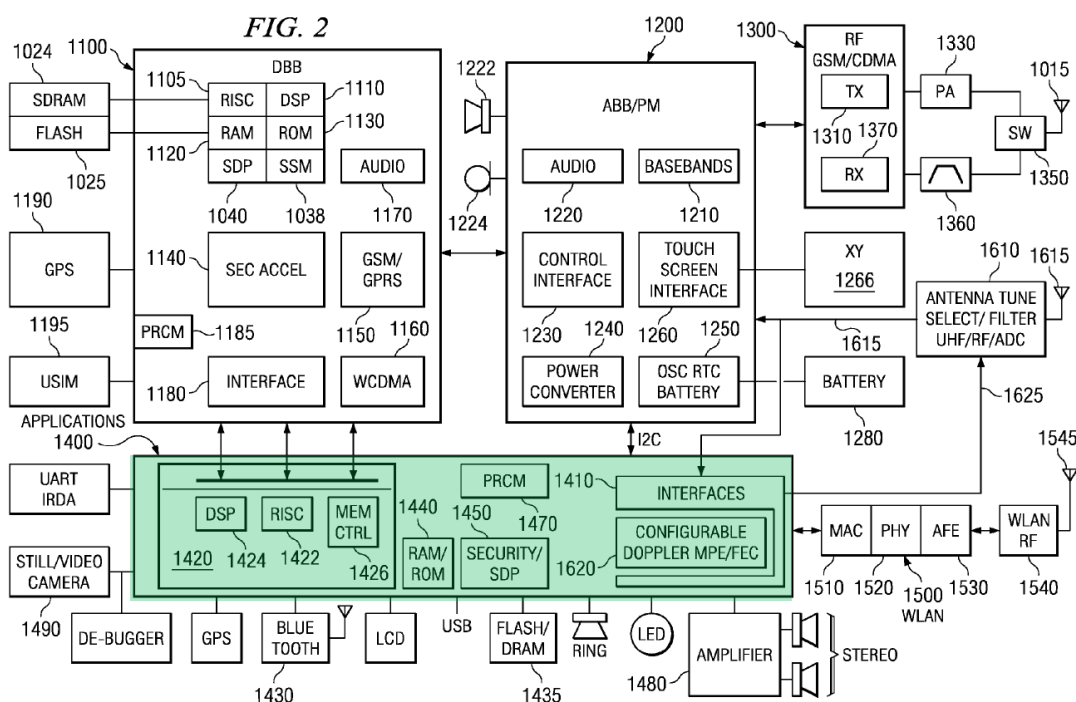
Id., 4:8–15 (emphasis added).

C. Titiano (Ex-1007)

42. Titiano was filed December 11, 2007, and published July 3, 2008. Ex-1006, code (43). Titiano qualifies as prior art under at least pre-AIA 35 U.S.C. § 102(a), (b), and (e).

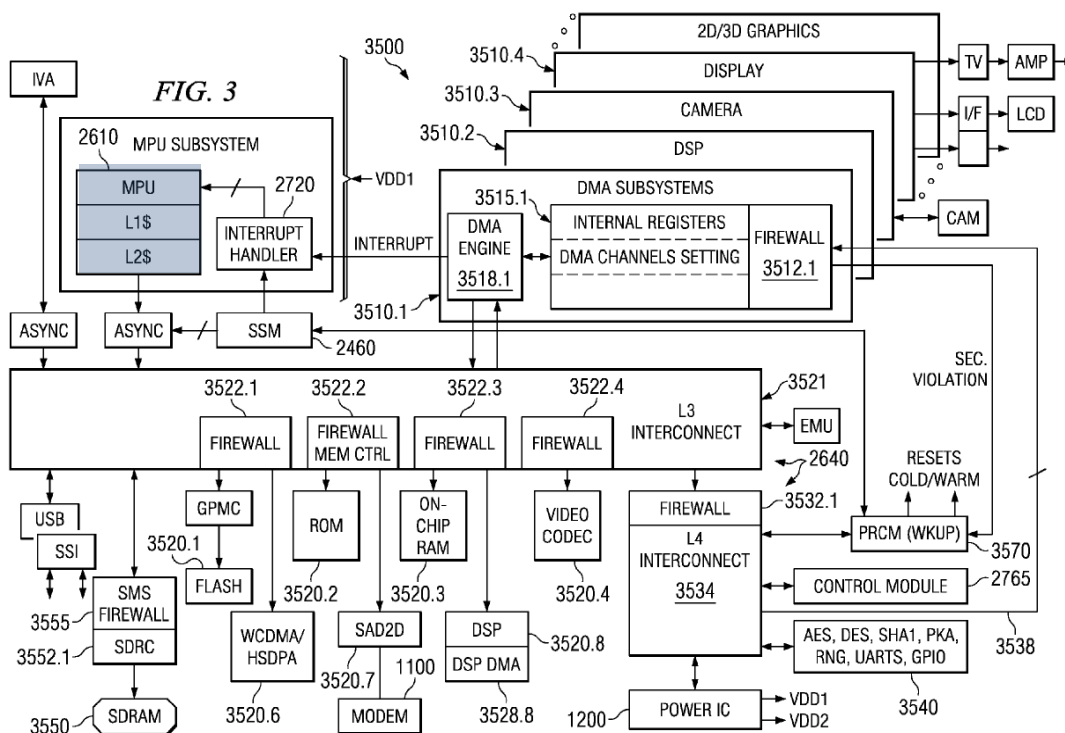
43. Titiano is directed “to conserving power in non-uniform cache access (NUCA) systems.” Ex-1007 ¶1. Some embodiments of Titiano describe “an electronic circuit includ[ing] processors operable to make respective voltage requests and a power management circuit having a controllable supply voltage output coupled to the processors.” *Id.* ¶11. The electronic circuit in Titiano further has “a voting circuit responsive to the voltage requests and operable to automatically establish a function of the respective voltage requests to control the controllable supply voltage output.” *Id.*

44. As annotated in Figure 2, “provided on chip 1400 is an applications processing section 1420 which includes a RISC processor 1422 (such as MIPS core(s), ARM core(s), or other suitable processor), a digital signal processor (DSP) 1424 such as from the TMS320C55x™ DSP generation and/or the TMS320C6x™ DSP generation from Texas Instruments Incorporated or other digital signal processor(s).” *Id.* ¶63. Further, in Figure 3, “a system 3500 has an MPU subsystem, an IVA subsystem, and DMA subsystems 3510.i. The MPU subsystem suitably has one or more processors with CPUs such as RISC or CISC processors 2610, and having superscalar processor pipeline(s) with L1 and L2 caches.” *Id.* ¶76.



45. Figure 3 shows the MPU subsystem of Figure 2, including “one or

more processors with CPUs such as RISC or CISC processors 2610, and having superscalar processor pipeline(s) with L1 and L2 caches.” *Id.* ¶¶76.



46. Titiano describes using Dynamic Voltage and Frequency Scaling (DVFS) to dynamically adjust the voltage and frequency of the cores to reduce power consumption. *Id.* ¶¶107, 109. As illustrated in Figure 8, the SoC Voltage Controller receives voltage requests from the CPU cores and selects the highest of the voltage requests as the “voted voltage,” which it uses to update the voltage setting for the cores.

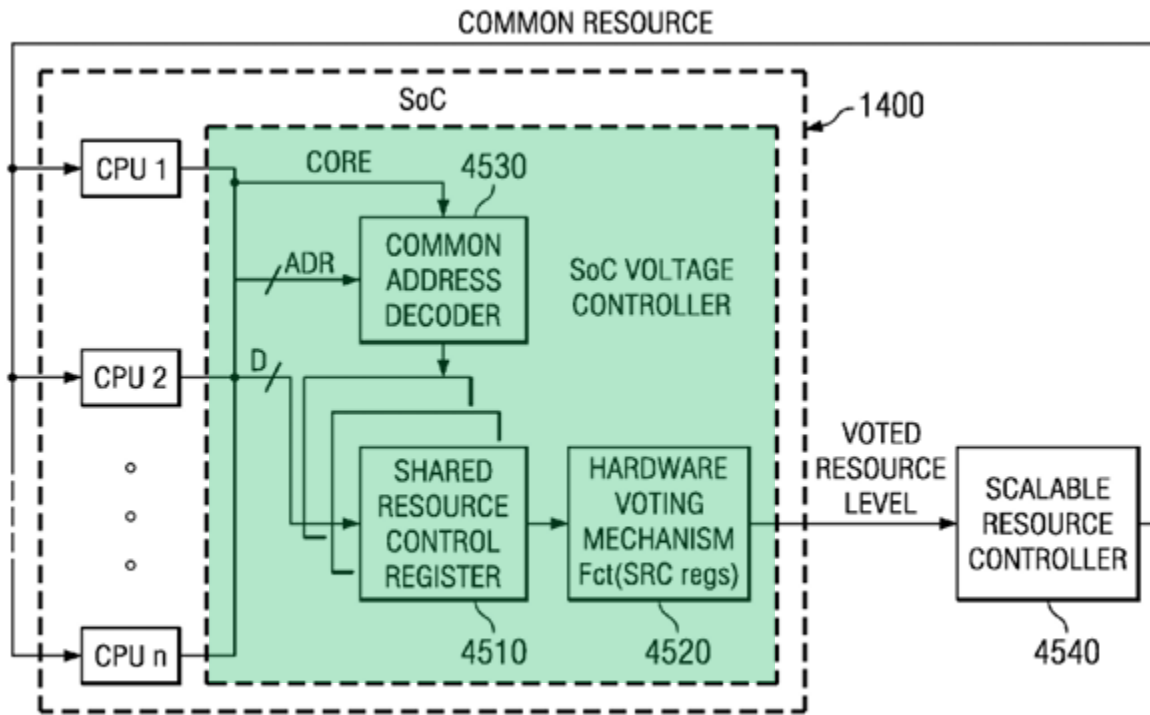


FIG. 8

47. Titiano also describes Dynamic Power Switching (DPS) and Adaptive Voltage Scaling (AVS) as additional features that may be combined with DVFS and used to manage power and performance of the multi-core processor. *Id.* ¶106. DPS is used to switch the cores to a sleep or low-power standby mode when they are not active. *Id.* ¶114. AVS is used to adjust the power supply voltage based, in part, on “the temperature induced current performance of the device.” *Id.* ¶174.

D. Hansquine (Ex-1008)

48. Hansquine was filed February 8, 2010, and published January 6, 2011. Hansquine qualifies as prior art under pre-AIA 35 U.S.C. § 102(a) and (e).

49. Hansquine is directed to “adaptive voltage scalers (AVSs), AVS

systems, and related circuits and methods.” Ex-1008 ¶8. Hansquine’s AVS system is “configured to adaptively adjust voltage levels powering a functional circuit(s) based on target operating frequencies and delay variation conditions to avoid or reduce voltage margin.” *Id.*, code (57). To implement AVS, Hansquine discloses a two-dimensional AVS table that “can be configured to store voltage levels for various operating frequencies of a functional circuit(s) to avoid or reduce voltage margin.” *Id.* ¶9.

AVS Frequency/Voltage Level Table (140)

142

	T (°C)		f ₁	f ₂	...	f _N
B ₁ →	80	60	1.2V	1.18V	...	
B ₂ →	60	40	1.1V	1.08V	...	
B ₃ →	40	20	1.0V	.99V	...	
B ₄ →	20	0	⋮	⋮	...	⋮
B _N →	⋮	⋮	.95V	.94V	...	

144

$f_1 > f_2 > f_N$

FIG. 13

50. As shown in Figure 13 above, Hansquine teaches that “a voltage level is retrieved from the database corresponding to the target operating frequency ... and a temperature level of the functional circuit(s).” *Id.*, code (57).

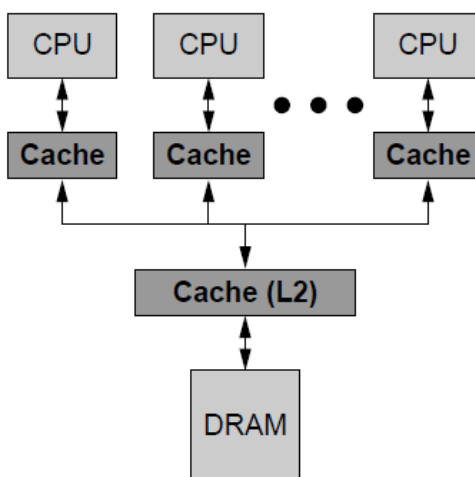
VI. TECHNOLOGICAL BACKGROUND

A. It was well-known to use an interconnect to couple cores in a multi-core processor to a shared cache memory.

51. The concept of a cache memory to store recently used data elements has been known and utilized since the earliest computers. As more complex systems were developed, it became common to utilize multiple levels of caching (referred to as “L1”, “L2”, etc.) in what became known as a “memory hierarchy.” Smaller and faster caches are closer to the processor, and larger and slower caches are farther away. Multiple levels of caching provide enhanced speed for most applications compared to a single level. As systems were developed that employed multiple processors that shared a single address space, it became common for each processor to have a private L1 cache and to share an L2 cache with other processors. Shared L2 caches are more efficient than private L2 caches but introduce the effect of interference between workloads operating on different processors.

52. A discussion of multiprocessor systems sharing L2 caches can be found in Chapter 1 of the well-known textbook: “Memory Systems, Cache, DRAM, Disk,” by Bruce Jacob, Spencer Ng and David Wang, published in 2008. This chapter discusses cache memory systems in detail and also applies it to multiprocessors. They illustrate a multiprocessor memory hierarchy and state that:

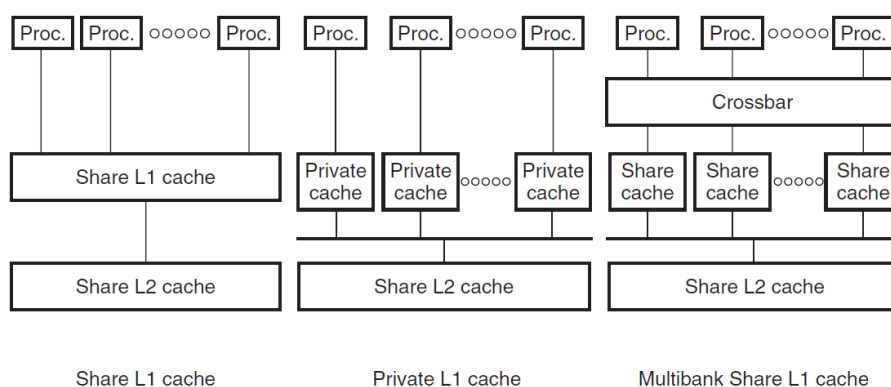
“Figure 1.1(d) shows a typical cache hierarchy for a multiprocessor organization. Each processor core has its own private cache, and the last-level cache before the backing store (DRAM) is shared by all cores.” Ex-1009at 60.



(d) multi-level cache hierarchy

53. In this figure, the lines connecting the L2 cache with the private caches represent an interconnect that couples the CPUs and the L2 cache.

54. Also, the well-known textbook “Multiprocessor Systems-on-Chips,” edited by Ahmed Amine Jerraya and Wayne Wolf and published in 2005, provides an introduction to multiprocessor chips, including how they employ memory hierarchies. In Chapter 2 of the book, entitled “Techniques for Designing Energy-Aware MPSoCs” by Mary Jane Irwin, Luca Benini, Narayanan Vijaykrishnan, and Mahmut Kandemir, various cache and processor organizations are described and illustrated: “A common characteristic of these configurations is that they all have a shared L2 cache.” Ex-1010 at 31.



2-8
FIGURE Different cache organizations: multiported shared cache, private caches, and CCC.

55. In this figure, the horizontal lines indicate buses that are used to interconnect the processors with the L2 cache. Also, with respect to the “CCC” organization, the authors state: “For performance reasons, a CCC architecture would need a crossbar interconnect between the processors and L1 cache banks. Compared with a bus-based architecture, the crossbar-based architecture provides simultaneous accesses to multiple cache banks (which, in turn, helps to improve performance).” *Id.* at 31. In summary, before the priority date of the ’197 Patent it was well-known to use an interconnect to couple cores in a multi-core processor to a shared cache memory.

B. It was well-known to monitor the workload of cores in a multi-core processor.

56. Since the earliest processor designs it has been appreciated that different workload types have different operational characteristics. Early processor designers distinguished between what were known as “scientific” workloads

(which are computationally intense with predictable control flow) and what were known as “general purpose” workloads (which require less computation and have control flow that is less predictable). These different characteristics led processor designers to measure and monitor workload characteristics to evaluate processors both for the purpose of optimizing design and to control run-time configuration. One way to monitor run-time behavior of a workload is to use what known as “performance counters,” which are registers built into a processor to count various events that are occurring. A two-part article by Brinkley Sprunt in 2002, entitled “The Basics Of Performance-monitoring Hardware; Pentium 4 Performance-Monitoring Features,” explains performance counters and surveys the state of the art in performance monitoring hardware. Ex. 1011 at 64. Sprunt explains that: “Most modern, high-performance processors have special, on-chip hardware that monitors processor performance. Data collected by this hardware provides performance information on applications, the operating system, and the processor.” *Id.*

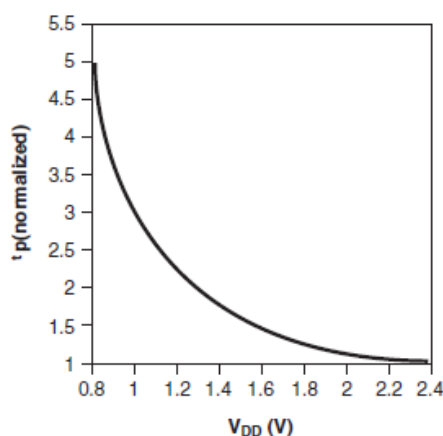
57. Performance-monitoring hardware has been incorporated into processor based systems at both the CPU level and at the system level. As processors have evolved into multi-core designs, it has become common to monitor workload performance for each core in a multi-core processor separately. For example, in the 2009 paper “Accuracy of Performance Counter

Measurements,” authors Dmitrijs Zapanuks, Milan Jovic and Matthias Hauswirth analyze performance counters on multi-core processors and note that: “On multi-core processors each core usually contains its own set of counters.” Ex-1012 at 2. Paul West also provides an extensive analysis of multi-core performance monitoring in his 2008 M.S. Thesis “Core Monitors: Monitoring Performance In Multicore Processors.” Ex-1013. West notes that: “Performance counters are becoming more complex as multi-core systems are becoming more wide spread” and that “watching memory traffic through a shared cache can determine if a program on a certain CPU is memory-bound.” *Id.* at viii.

58. The use of workload characteristics is also explored in the 2009 M.S. Thesis “Workload Based Lookup Table for Minimal Power Operation Under Supply and Body Bias Control” by K. Sreejith. Ex-1014. Sreejith notes that: “[voltage] values for minimal power consumption varies with the workload of the circuit” and proposes “workload indexed lookup table to provide the optimum [voltage] supply ... to minimize the total power consumption of the circuit under different operating conditions.” *Id.* at 4. Sreejith defines “workload” as “the ratio of the number of cycles for which the block (unit) is active to the total number of cycles” and proposes the use of this metric “as a proxy for the activity of a block.” *Id.* at 9. In summary, before the priority date of the ’197 Patent, it was well-known to monitor the workload of cores in a multi-core processor.

C. It was well-known to control the frequency and voltage of cores to optimize for power or performance and use a table to store voltage and frequency values.

59. It has been well-understood for decades that there is a relationship between the operating voltage, the frequency, and the power consumption of digital systems implemented using semiconductors. This is because of a feature of MOS transistors in which the “delay,” or the time to switch from one state to the other, is reduced as the voltage increases. For example, Irwin et al. note that: “the best knob for controlling power is setting the supply voltage appropriately to meet the computational load requirements. Although lowering the supply voltage has a quadratic impact on active energy, it decreases systems performance since it increases gate delay, as shown in Figure 2-4.” Ex-1010 at 24–25.



2-4
FIGURE Delay as a function of V_{DD} .

60. Irwin et al. discuss multiple techniques that have been used to control power consumption, including “dynamic voltage scaling (DVS)” and “dynamic

frequency scaling (DFS)”:

	Constant Throughput/Latency		Variable Throughput/Latency
	Design Time	Sleep Mode	Run Time
Active (Dynamic) [C V ₂ f]	Logic design Trans sizing Multiple V _{dd} 's	Clock gating	DFS DVS DTM
Standby (Leakage) [V I _{off}]	Stack effect Multiple V _T 's Multiple t _{ox} 's	Sleep trans Multi-V _{dd} Variable V _T Input control	Variable V _T

2-3
FIGURE

Processor power design space. DFS, dynamic frequency scaling; DVS, dynamic voltage scaling; DTM, dynamic thermal management.

Id. at 24.

61. Irwin et al. explain that: “The most popular of the techniques for reducing active power consumption at run time is DVS combined with dynamic frequency scaling (DFS). Most embedded and mobile processors contain this feature (triggered by thermal on-chip sensors when thermal limits are being approached or by the run-time system when the CPU load changes). ... Future MPSoCs in which all (or selected) processing cores are DFS + DVS capable would require each to have its own converter and PLL (or ring oscillator) as well as system software capable of determining the optimum setting for each core, taking into account the computational load.” *Id.* at 24–25.

62. Also, in a 2004 White Paper (“Intel SpeedStep”), Intel introduced what they referred to as SpeedStep® technology that implemented a set of performance states with different frequency and voltage settings for each state. Ex-

1015 at 4. Intel SpeedStep illustrates the voltage and frequency settings using a table of values:

Table 1.1 Supported Performance States for the Intel® Pentium® M Processor at 1.6GHz¹

Frequency	Voltage
1.6 GHz (HFM)	1.484 V
1.4 GHz	1.420 V
1.2 GHz	1.276 V
1.0 GHz	1.164 V
800 MHz	1.036 V
600 MHz (LFM)	0.956 V

Id. Higher values for frequency and voltage correspond with high performance demands (i.e., heavy power consumption), while lower values for frequency and voltage correspond with lower performance demands (i.e., power saving modes).

63. Intel SpeedStep also states that: “The top and bottom modes are commonly known as high frequency mode (HFM) and low frequency mode (LFM). These frequency and voltage operating points are stored within a read-only processor model specific register (MSR). This MSR ensures BIOS will not allow transitions to invalid states above the HFM maximum or below the LFM minimum. ***The other four operating points are stored within BIOS code, as a drop in voltage table*** provided by Intel to BIOS vendors.” *Id.* (emphasis added).

64. In summary, before the priority date of the '197 Patent, it was well-known to control the frequency and voltage of cores to optimize for power or performance and use a table to store voltage and frequency values.

VII. DETAILED EXPLANATION OF THE UNPATENTABILITY GROUNDS

65. The '197 Patent contains 20 claims. Similar claim elements are discussed together in the below element-by-element analysis. Meaningful differences are individually identified and discussed even if not explicitly identified in the section headings. As explained below, the subject matter of the challenged claims is disclosed by the prior art.

A. Grounds 1 and 2: Claims 1–3 and 19–20 Are Obvious Over Naveh and Obvious Over Naveh in view of Saeed

1. A POSITA would have been motivated to combine Naveh with Saeed

66. In addition to the reasons set out below, a POSITA would have been motivated to combine Naveh with Saeed's teachings, and would have had a reasonable expectation of success in doing so, because they are from the same field of endeavor and relate to the same well-known technologies. Naveh and Saeed provide complementary techniques for power management of multi-core processors. Ex-1005 ¶¶4–7, 19, 23–26, 29, 41; Ex-1006 at 2:47–3:7. A POSITA seeking to increase power management and efficiency in a multi-core processor would have known of and consulted each of these proposed techniques. More specifically, both Naveh and Saeed are concerned with reducing power consumption and improving performance of multi-core processors. Naveh explains

that “as the trend toward advanced processors with more transistors and higher frequencies continues to grow, computer designers and manufacturers are often faced with corresponding increases in power consumption,” which in turn “result[s] in increased leakage power.” Ex-1005 ¶4. Naveh stresses that this is the case “[p]articularly in mobile computing environments, [where] increased power consumption can lead to overheating, which may negatively affect performance, and can significantly reduce battery life.” *Id.* Similarly, Saeed explains that when “tasks are performed in parallel with one another on separate processing cores, ... the [multi-threaded] MT workload also consumes more power.” Ex-1006, 1:19–23. Further, “[i]n battery powered computer systems such as a laptop or notebook, the higher the power consumption, the faster the remaining battery life is depreciated.” *Id.*, 1:124–26.

67. Moreover, Naveh teaches techniques for optimizing performance or power in multi-core processors. Ex-1005 ¶¶22–23, 26–27, 33, 41. For example, Naveh teaches using a policy or mode to determine power settings for multiple cores based on operating requests. One such policy is a minimum or power-based policy, which implements the lowest operating request across all the cores. *Id.* ¶27. Another such policy is a maximum or performance-based policy, which implements the highest operating request across all the cores. *Id.* ¶33.

68. Saeed teaches using workload as a parameter to determine a mode or

policy for a processor. *See* Ex-1006 at 4:25–26 (“the modes of operation are determined by the level of multi-threadedness”); *id.*, 2:10–11 (“the workload may be defined by the level of multi-threadedness”). Saeed also teaches using the selected mode to update power settings such as frequency and voltage. *Id.*, 3:8–22. For example, “[i]f the system is placed in the maximum frequency mode 301, all CPU cores of the computer system are set to the maximum operating frequency 311.” *Id.*, 3:26–29; *see also* Fig. 3. A POSITA would understand from these teachings that selecting one mode over another, with the goal of optimizing for power or performance, should depend at least in part on the workload of the system. *See supra* Section VI.B. Saeed provides additional well-known teachings of adjusting settings of power management features based on the workload of a processor. Ex-1006, 2:47–3:7. Thus, a POSITA would have been motivated to modify Naveh to select a global operating mode or policy based on the workload of the processor. Such modification would lead to a more adaptive processor and improve efficiency, which lowers operating costs and extends useful life of the products. *See* Ex-1005 ¶¶9, 70, 106, 110, 252–53; Ex-1006, 1:15–25, 3:3–5, 3:28–30.

69. A POSITA would have had a reasonable expectation of success in combining the teachings of Naveh and Saeed because each purports to address these issues in similar and predictable ways—for example, by using well-known

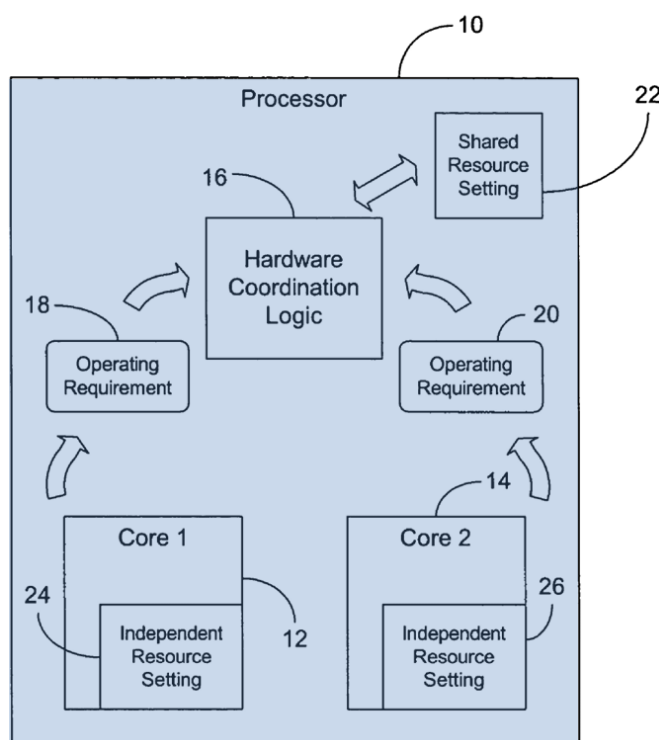
techniques, such as monitoring the workload of applications and setting a mode or policy based on the workload. Naveh focuses on power management based on the performance needs of the cores, and Saeed focuses on power management based on the workload of the processor. A POSITA would have understood that they could have combined these teachings with no change in their respective functions because they teach substantially similar systems.

2. Independent Claim 1

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

70. To the extent that the preamble of Claim 1 is limiting, Naveh discloses the preamble.

71. Naveh is directed to “coordinating power management activities in multi-core processors.” Ex-1005 ¶2. Further, as shown in annotated Figure 1 below, Naveh provides that the multi-core “processor 10 has a number of features/resources that enable a significant reduction in power. For example, the processor 10 may have a feature that is able to scale down the processor clock frequency and power supply voltage (e.g., core voltage) when high performance is either not needed or not desired.” *Id.* ¶18; Fig. 1.

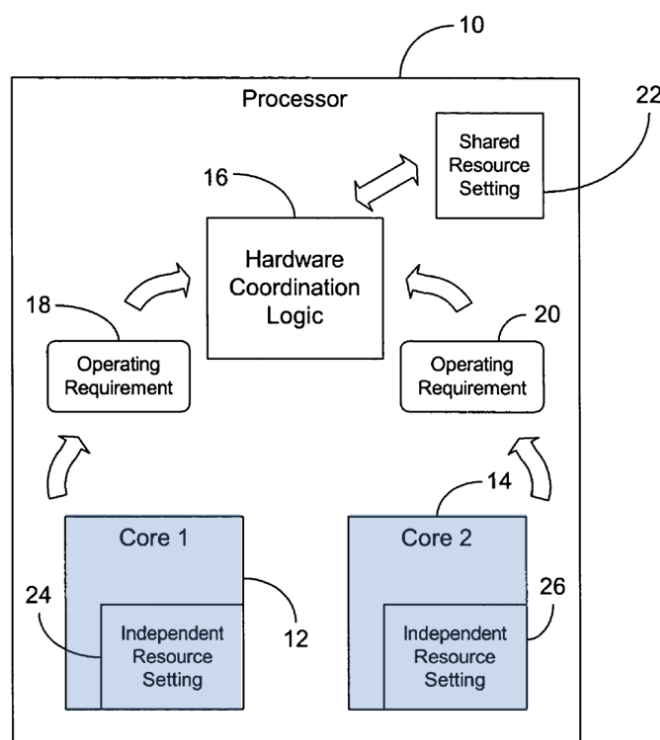


72. Naveh’s processor has a number of applications, including “a mobile computing system such as a notebook personal computer (PC), a personal digital assistant (PDA), a wireless ‘smart’ phone, etc., in which power consumption and overheating are issues of particular concern.” *Id.* ¶36.

b. Element 1[a]: a plurality of cores;

73. Naveh discloses Element 1[a]. Cores 12 and 14 in Figure 1 correspond to the claimed “plurality of cores.” Naveh is directed to “coordinating power management activities in multi-core processors,” and discloses “[s]ystems and methods of managing power ... [by] issuing a first operating requirement from a first processor core and issuing a second operating requirement from a second

processor core.” Ex-1005, code (57), ¶2. For example, Naveh’s Figure 1 (annotated below) “shows a processor 10 [(blue)] having a first core 12, a second core 14 [(blue)] and hardware coordination logic 16 operatively coupled to the first core 12 and the second core 14”. *Id.* ¶17; *see also id.* ¶22 (“In the illustrated embodiment, the first core 12 issues a first operating requirement 18 and the second core 14 issues a second operating requirement 20.”).



74. Naveh also notes that “while a dual-core configuration is shown here, the number of cores in the processor 10 can readily be increased without parting from the spirit and scope of the embodiments of the invention.” *Id.* ¶17. Thus, a POSITA would have understood that the power management concepts and

techniques described in Naveh could be implemented with respect to any number of cores.

c. Element 1[b]: a cache memory;

75. Naveh discloses Element 1[b]. Naveh discloses that “[p]ower management can also be enhanced by scaling the frequency of the bus that the processor uses to communicate with other components such as chipsets, *memory subsystems*, input/output (I/O) devices, etc.” Ex-1005 ¶5 (emphasis added). A POSITA would have understood to include a shared L2 cache in Naveh’s memory subsystems. As noted above, multi-core processors with shared L2 caches were common. *Supra* Section VI.A. A shared L2 cache was a well-known element used within a memory hierarchy to complement L1 caches used by each core. *Id.* As noted above, a shared L2 cache avoids the problem of inefficient utilization if each core were to have both private L1 and L2 caches. *Id.* Thus, it would have been obvious to implement a shared L2 cache in the memory subsystem of Naveh’s processor.

d. Element 1[c]: an interconnect to couple the plurality of cores to the cache memory;

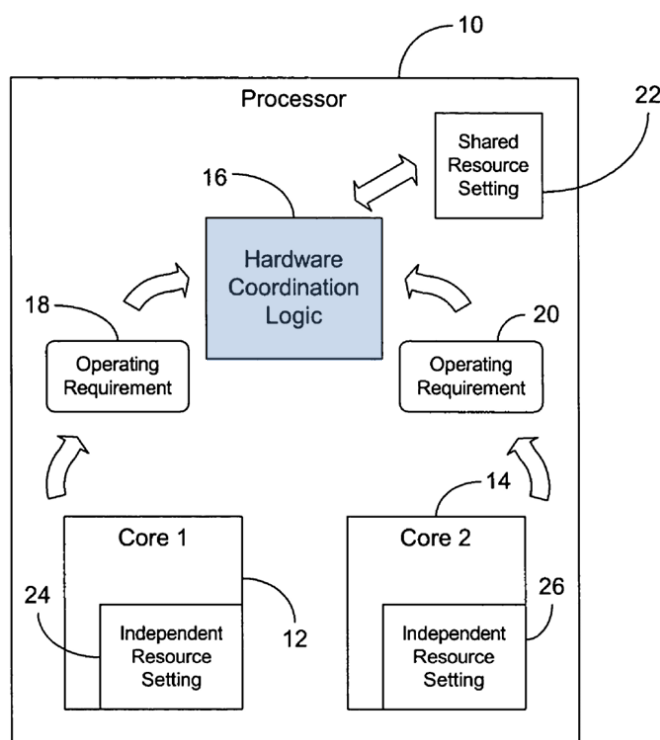
76. Naveh teaches Element 1[c]. As discussed above with reference to Elements 1[a] and 1[b], Naveh discloses a processor with a plurality of cores and a cache memory. *See supra* Sections VII.A.2.b and VII.A.2.c. The cores must be

able to access the cache memory, and thus it would have been obvious to a POSITA that the processor in Naveh would include an interconnect to couple the plurality of cores to the cache memory. *See supra* Section VI.A. For example, Naveh discloses that processor 10 of Figure 1 may include a bus (even though not illustrated in Figure 1). Ex-1005 ¶18. A POSITA would have understood that an interconnect, such as a bus, could be part of the interconnect that couples the plurality of cores to the cache memory.

77. For at least these reasons, Naveh discloses or renders obvious an interconnect to couple the plurality of cores to the cache memory.

e. Element 1[d]: a power controller to control a plurality of power management features of the processor;

78. Naveh discloses Element 1[d]. With respect to Figure 1 (annotated below), Naveh teaches that “coordination logic 16 can coordinate the shared resource setting 22 with the operating requirements 18, 20.” Ex-1005 ¶22; *see also id.* ¶17 (“FIG. 1 shows a processor 10 having a first core 12, a second core 14 and hardware coordination logic 16 operatively coupled to the first core 12 and the second core 14.”).



79. Naveh further discloses that coordination logic 16 is implemented by hardware. *See, e.g., id.* ¶28 (“[B]y implementing the coordination logic in hardware, fast OS response times in order to accommodate rapid state changes are not imposed on the system.”); *id.* ¶37 (“For example, one or more portions of the method 96 can be embodied in fixed functionality hardware, an application specific integrated circuit (ASIC), in a set of microcode instructions to be stored on a machine readable medium, or any combination thereof.”); *id.* ¶41 (“[H]ardware coordination is relatively fast and obviates response time concerns associated with OS-coordinated power management.”).

80. Naveh teaches that coordination logic 16 controls power management

settings shared by cores 12 and 14, which are called shared resource settings 22.

Ex-1005 ¶20, 22. Naveh identifies clock frequency and power supply voltage as

examples of shared resource settings 22. *Id.* ¶¶20, 31. By controlling shared

resource settings 22 (such as clock frequency and power supply voltage),

coordination logic 16 “control[s] a plurality of power management features of the processor,” as recited in Element 1[d].

- f. Element 1[e]: wherein the power controller includes a tuning circuit to [e1] receive a workload configuration input regarding a workload, [e2] receive a plurality of energy performance bias (EPB) values and determine a global EPB value based thereon, and [e3] update at least one setting of at least one of the plurality of power management features based on the workload configuration input and the global EPB value**

81. Naveh teaches Element 1[e]. The hardware coordination logic 16 in Naveh, which is implemented in hardware, corresponds to the power controller.

Ex-1005 ¶28. Naveh states, “[i]t should be noted that by implementing the coordination logic in hardware, fast OS response times in order to accommodate rapid State changes are not imposed on the system.” *Id.* Thus, the hardware portion of coordination logic 16 that performs the functions recited in Elements 1[e1], 1[e2] and 1[e3] corresponds to the claimed “tuning circuit.”

82. As for Elements 1[e1], 1[e2] and 1[e3], this paragraph presents a brief overview of Naveh’s teachings and the following subsections provide more

detailed analysis. According to Naveh, coordination logic 16 receives operating requirements 18 and 20 from cores 12 and 14, compares operating requirements 18 and 20, selects one of them as shared resource setting 22. Ex-1005 ¶¶26, 39.

Operating requirements 18 and 20 received from cores 12 and 14 correspond to “a plurality of energy performance bias (EPB) values,” and the operating requirement selected as shared resource setting 22 corresponds to “a global EPB value.” *See infra* Section VII.A.2.f.i–iii. Naveh further teaches that coordination logic 16 may select shared resource setting 22 based on a performance/maximum policy or power/minimum policy. Ex-1005 ¶23. In Naveh, the “workload configuration input” corresponds to the information that coordination logic 16 receives and uses to determine whether to apply a performance/maximum policy or power/minimum policy. *See infra* Section VII.A.2.f.i.

i. Element 1[e1]: a tuning circuit to receive a workload configuration input regarding a workload;

83. Naveh teaches Element 1[e1]. For “workload configuration input” in Elements 1[e1] and 1[e3], this Declaration presents two alternative Grounds, which are addressed separately below.

Ground 1 (Naveh renders Element 1[e1] obvious):

84. As summarized above and described in additional detail below in Element 1[e2], coordination logic 16 receive operating requirements 16 and 18

from cores 12 and 14. Ex-1005 ¶22. Coordination logic 16 compares operating requirements 16 and 18 and selects one of them to be shared resource setting 22. *Id.* ¶¶26–27. In one mode, coordination logic 16 selects the higher of the two operating requirements. *Id.* ¶¶26, 39. Naveh refers to this mode as a performance/maximum policy. *Id.* ¶¶23, 27. In another mode, coordination logic 16 selects the lower of the two operating requirements. *Id.* ¶27. Naveh refers to this mode as a power/minimum policy. *Id.* ¶¶23, 27.

85. Naveh also states that the mode or policy may change over time depending on “what is currently the most important factor [i.e., performance or power] for software to control.” *Id.* ¶23, 41. Thus, coordination logic 16 receives from software an indication of whether the performance/maximum policy or the power/minimum policy should be applied. This policy setting is a workload configuration input. A performance/maximum policy indicates that the current workload is of a type that needs to be processed quickly, and thus performance is more important than power for this workload. A power/minimum policy indicates that the current workload is of a type that does not need to be processed quickly, and thus power is more important than performance for this workload. Thus, the policy setting is a workload configuration input, and Naveh teaches that hardware coordination logic 16 receives this workload configuration input from the software.

86. To the extent Patent Owner argues that the performance/maximum or

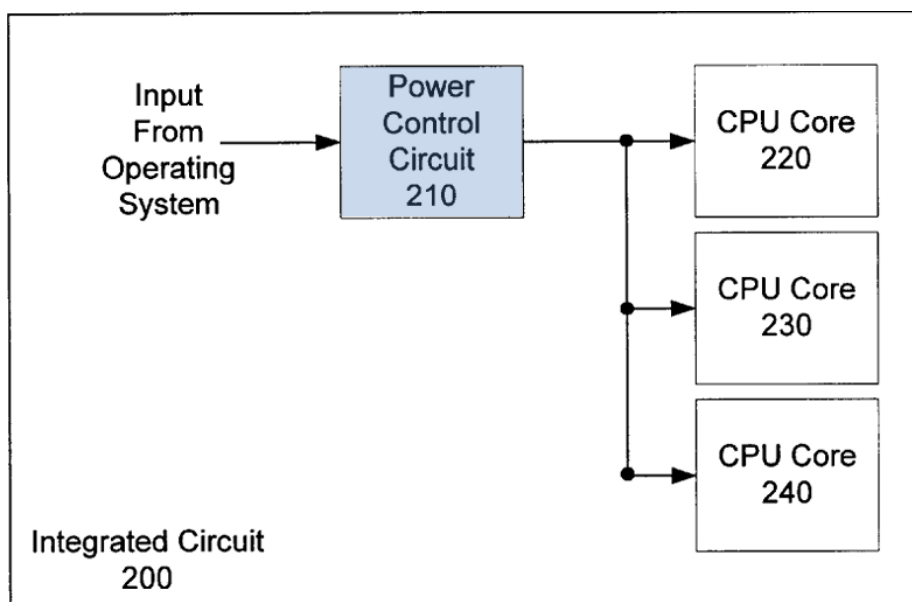
power/minimum policy setting is not a workload configuration input, it would have been obvious to a POSITA that hardware coordination logic 16 could also receive a workload coordination input to determine whether to apply a performance/maximum policy or a power/minimum policy. *See supra* Section VI.B. Naveh teaches that the policy may change over time depending on “what is currently the most important factor [i.e., performance or power] for software to control.” Ex-1005 ¶¶23, 41. Thus, it would have been obvious to a POSITA that the policy described in Naveh could be set based on a workload configuration input received from a user, the operating system, or the cores on the multi-core processor. *See supra* Section VI.B. Thus, it would have been obvious that the tuning circuit of Naveh receives a “workload configuration input regarding a workload.”

Ground 2 (Naveh in view of Saeed renders Element 1[e1] obvious):

87. The Naveh-Saeed combination further supports that it would have been obvious to modify Naveh to receive a workload configuration input to determine whether to apply a performance/maximum policy or a power/minimum policy.

88. Saeed discloses an integrated circuit with a power control circuit for controlling the frequency and voltage of a plurality of cores. For example, Saeed’s Figure 2 (annotated below) “depicts an integrated circuit (IC) 200 that has a power

control circuit for deactivating or lowering a frequency or voltage of a CPU core if a calculated workload is less than a defined threshold.” Ex-1006, 3:8–11. In this figure, power control circuit 210 is coupled to three CPU’s cores 220, 230, and 240. *Id.*, 3:11–13.



In Saeed, power control circuit 210 is a power controller including a tuning circuit.

89. Like Naveh, Saeed recognizes that there is a tradeoff between performance and power consumption (*id.*, 1:20–26, 2:1–4) and that the voltage and frequency applied to a plurality of cores affect performance and power consumption (*id.*, 2:63–3:5). Saeed further describes making a decision about performance and power consumption based on “workload.” *See id.*, 3:8–11 (“deactivating or lowering a frequency or voltage of a CPU core if a calculated workload is less than a defined threshold”); *id.*, 4:35–37. Saeed also describes

placing the system into different power/performance modes based on “workload.”

Id., 4:24–35. In particular, Saeed states that the OS may determine the workload.

Id., 3:65–66. “The power control circuit 210 receives as input a measured workload from the operating system.” *Id.*, 3:13–15. The power control circuit then selects a mode of operation based on whether the workload is above or below a threshold. *Id.*, 4:24–35.

90. It would have been obvious to a POSITA to apply the teachings of Saeed to hardware coordination logic 16 of Naveh. *See also supra* Section VII.A.1. Specifically, it would have been obvious for coordination logic 16 of Naveh “to receive a workload configuration input regarding a workload” so that it could use the workload configuration input to determine the mode of operation, as taught by Saeed. *See supra* Section VII.A.1. Hardware coordination logic 16 could use the workload configuration input to decide whether to apply a performance/maximum or a power/minimum policy. For example, if the workload configuration input (e.g., level of multi-threadedness) was above a threshold, as Saeed teaches ((Ex-1006, 2:27–32), coordination logic 16 could apply a performance/maximum policy and select the higher the two cores’ operating requirements to be shared resource setting 22, and if the workload configuration input was below the threshold, as Saeed teaches (*id.*), coordination logic 16 could apply a power/minimum policy and select the lower of the two cores’ operating requirements as shared resource

setting 22. A POSITA would have understood that the receipt and use of a workload configuration input would allow coordination logic 16 to apply a policy appropriate for the demands of applications running on the processor. This would result in a better balance of performance and power, improving the performance and power efficiency of the processor. *See supra* Section VII.A.1.

ii. Element 1[e2]: a tuning circuit to ... receive a plurality of energy performance bias (EPB) values and determine a global EPB value based thereon;

91. Naveh discloses Element 1[e2]. As stated above, the portion of coordination logic 16 that performs the functions recited in Elements 1[e1], 1[e2] and 1[e3] is the claimed “tuning circuit.” *See supra* Section VII.A.2.f. According to Naveh, coordination logic 16 receives operating requirements 18 and 20 from cores 12 and 14, compares operating requirements 18 and 20, and selects one of them as shared resource setting 22. Ex-1005 ¶¶26, 27, 39. Operating requirements 18 and 20 are “a plurality of energy performance bias (EPB) values,” and the one selected as shared resource setting 22 is “a global EPB value.” Thus, Naveh teaches “a tuning circuit ... to receive a plurality of energy performance bias (EPB) values and determine a global EPB value based thereon.”

92. With reference to Figure 1, Naveh teaches that coordination logic 16 receives operating requirements 18 and 20 from cores 12 and 14, and coordination

logic 16 determines shared resource setting 22 based on operating requirements 18 and 20. Ex-1005 ¶22.

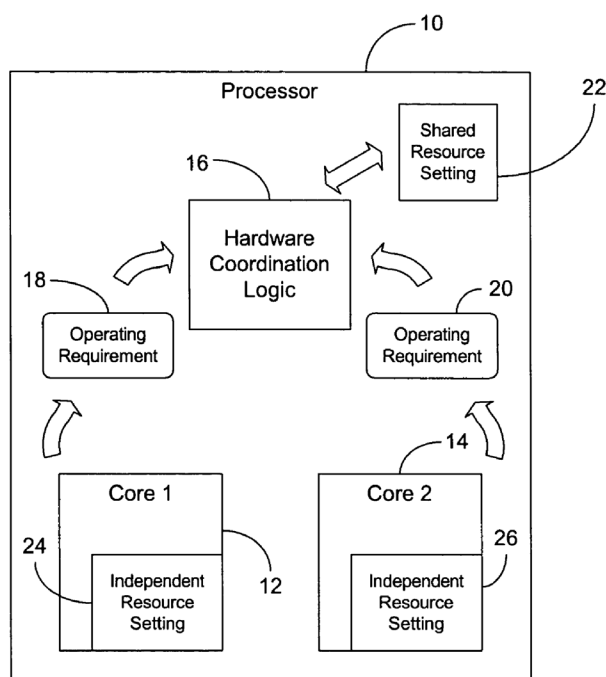


FIG. 1

Just as the ‘197 Patent states that tuning circuit 100 may “receive incoming EPB values from multiple cores of a multicore processor” (Ex-1001, 4:59–60), Naveh describes coordination logic 16 receiving operating requirements 18 and 20 from cores 12 and 14. Ex-1005 ¶22. Naveh states that after the operating requirements are “received by the coordination logic, they can be compared to one another for selection of an appropriate *global* setting.” *Id.* ¶31 (emphasis added). In one mode, coordination logic 16 selects the higher of the two operating requirements. *See id.* ¶26 (“It should also be noted that, the coordination logic 16 could simply compare

the operating requirements 18, 20 and select the higher of the two. The coordination logic 16 is therefore able to function as a ‘maximum detector’ for operating requirements and/or target values of the shared resource setting 22.”). Naveh refers to this mode as a “maximum” or “performance” policy. *Id.* ¶¶23, 27. In another mode, coordination logic 16 selects the lower of the two operating requirements. *Id.* ¶27. Naveh refers to this mode as a “minimum” or “power” policy. *Id.* ¶¶23, 27.

93. Figure 7 provides an example of coordination logic 16 implementing the performance/maximum policy. Coordination logic 16 receives first operating requirement 18 from first core 12 at block 98 and second operating requirement 20 from second core 14 at block 96. Coordination logic 16 compares the first and second operating requirements at blocks 112 and 114 and selects the greater of the operating requirements as shared resource setting 22 at blocks 116 and 118. *Id.* ¶39.

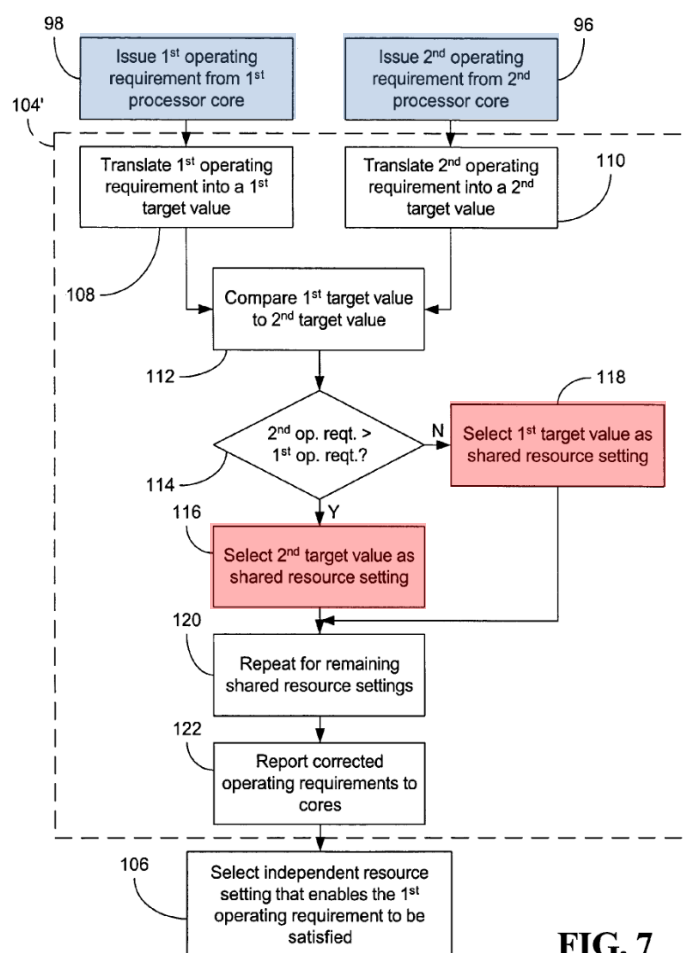


FIG. 7

94. Operating requirements 18 and 20 and shared resource setting 22 represent tradeoffs between power and performance and, thus, are EPB values. According to the '197 Patent, energy performance bias (EPB) values represent a tradeoff between power and performance. Ex-1001, 2:34–38 (“a power/performance tradeoff preference”), *id.*, 3:49–51 (“Each EPB value can be mapped to a corresponding level of tradeoff between power and performance.”); *id.*, 4:58–64 (“power management tuning circuit 100 can ... use these [EPB] values to determine an appropriate tradeoff between power and performance policies”).

95. Naveh also describes a tradeoff between power and performance as in the '197 Patent. Specifically, Naveh describes saving power by scaling down the frequency and voltage of the cores when performance is not needed or desired:

“For example, the processor 10 may have a feature that is able to scale down the processor clock frequency and power supply voltage (e.g., core voltage) when high performance is either not needed or not desired.” Ex-1005 ¶18; *see id.* ¶¶23, 41.

Because cores 12 and 14 share a clock frequency and power supply voltage, “the clock frequency and power supply voltage settings could be viewed as a shared resource setting 22.” Ex-1005 ¶20.

96. Figure 3 of Naveh illustrates this power-performance tradeoff.

34

Combined Optimal State				Shared Feature SpeedStep® Control				Independent Feature TM1 Control		
Pindex	Perf	Abs-Pwr	Pwr%	Frequency	Vcc	Pwr%	Perf	Throttle%	Pwr%	Perf
0	1	100	100%	1700	1.233	100%	100%	0	1	100%
1	0.97	84	84%	1600	1.183	84%	97%	0	1	100%
2	0.94	68	68%	1500	1.133	68%	94%	0	1	100%
3	0.91	54	54%	1400	1.083	54%	91%	0	1	100%
4	0.88	41	41%	1300	1.008	41%	88%	0	1	100%
5	0.85	28	28%	1200	0.933	28%	85%	0	1	100%
6	0.82	16	16%	1100	0.85	16%	82%	0	1	100%
7	0.79	15	15%	1000	0.85	15%	79%	0	1	100%
8	0.76	13	13%	900	0.85	13%	76%	0	1	100%
9	0.73	12	12%	800	0.85	12%	73%	0	1	100%
10	0.7	10	10%	700	0.85	10%	70%	0	1	100%
11	0.67	9	9%	600	0.85	9%	67%	0	1	100%
12	0.58625	8	8%	600	0.85	9%	67%	0.125	0.9063	88%
13	0.5025	7	7%	600	0.85	9%	67%	0.25	0.8125	75%
14	0.41875	6	6%	600	0.85	9%	67%	0.375	0.7188	63%
15	0.335	6	6%	600	0.85	9%	67%	0.5	0.625	50%
16	0.1675	4	4%	600	0.85	9%	67%	0.75	0.4375	25%

50 52 42 44 36 38 40

FIG. 3

The higher frequencies and voltages in columns 36 and 38 correspond to higher performance optimization (as shown in the two columns labeled “Perf”) and correspond to less power optimization (as shown in the columns labeled “Abs-Pwr” and “Pwr%”). So, Naveh also recognizes a tradeoff between high performance and power optimization as in the ’197 Patent.

97. Operating requirements 18 and 20 and shared resource setting 22 correspond to particular rows in Figure 3 and, thus, represent a tradeoff between power and performance. Operating requirements 18 and 20 “identify a performance requirement such as a performance level or an index into a performance state transition table,” such as Figure 3. Ex-1005 ¶23. With reference to Figure 3, Naveh states:

In the illustrated example, the operating requirements can be defined as power requirements such as a combined optimal state absolute power (“Abs-Pwr”) 42 or a power percentage (“Pwr %”) 44. The operating requirements could also be defined as performance requirements such as a performance index (“Pindex”) 50 or a combined optimal state absolute performance (“Perf”) 52.

Id. ¶30. Naveh recognizes that coordination logic 16’s selection of one of operating requirements 16 and 18 as shared resource setting 22 represents a tradeoff between power and performance: “By enabling the hardware coordination logic 16 to function based on either a performance or a power policy, the processor 10 is able

to focus on what is currently the most important factor for software to control, while perhaps compromising the other.” *Id.* ¶23; *see also id.* ¶41.

98. Shared resource setting 22 is a “global” EPB value because it indicates an energy performance bias for both cores 18 and 20 on processor 10. Naveh describes selected resource setting 22 as a “global” setting. *Id.* ¶31 (“As the operating requirements and target values are received by the coordination logic, they can be compared to one another for selection of an appropriate **global** setting.” (emphasis added)); *id.* ¶30 (“To support relatively complex hardware coordination, it can be seen that shared resource settings may be integrated across various features and the actual operating requirements may be specified at a ‘**global**’ level.”(emphasis added)); *id.* ¶33 (“If the illustrated system is in the ‘max detect’ mode (i.e., Force_Min is not set), the first shared resource control block 64 compares the two operating requirements and determines that the appropriate **global** setting is 100% performance. Accordingly, the target frequency/voltage value of 1700 MHz/1.233V may be selected as the shared resource setting 68.” (emphasis added)).

99. Accordingly, Naveh discloses Element 1[e2].

iii. Element 1[e3]: a tuning circuit to ... update at least one setting of at least one of the plurality of power management features based on the workload configuration input and the global EPB value.

100. Naveh teaches Element 1[e3]. As explained above, the claimed “tuning circuit” in Naveh is the portion of coordination logic 16 that performs the functions recited in Elements 1[e1], 1[e2] and 1[e3], and the “global EPB value” is shared resource setting 22. *See supra* Section VII.A.2.f.

101. Coordination logic 16 updates a power management feature based on shared resource setting 22 (the global EPB value). For example, as discussed above for Element 1[e1], Naveh discloses that “clock frequency and power supply voltage settings could be viewed as a shared resource setting.” Ex-1005 ¶20; *see id.* ¶18. After coordination logic 16 selects the shared resource setting, it is “applied to the clock source and/or voltage source, instructing the shared resources to operate at the appropriate frequency and/or core voltage defined by the shared resource setting.” *Id.* ¶33. Thus, coordination logic 16 updates a power management feature (here, the frequency of the clock source and voltage of the voltage source) based on shared resource setting 22. *Id.*, Fig. 3.

102. As discussed above for Element 1[e1], the “workload configuration input” in Naveh corresponds to the information that coordination logic 16 receives and uses to determine whether to apply a performance/maximum policy or

power/minimum policy. *See supra* Section VII.A.2.f.i. Because this workload configuration input in Naveh affects the selection of shared resource setting 22 (i.e., whether coordination logic 16 picks the maximum or minimum of operating requirements 16 or 18), the updating of the power management feature is also based on the workload configuration input. Thus, Naveh teaches that coordination logic 16 “update[s] at least one setting of at least one of the plurality of power management features based on the workload configuration input and the global EPB value.”

3. Dependent Claims 2–3, 19–20

- a. Claim 2: The processor of Claim 1, wherein the power controller is to access a power-performance tuning table based on the workload configuration input and the global EPB value, and use information from the power-performance tuning table to update the at least one setting.**

103. Naveh teaches dependent claim 2. The Declaration presents two grounds for claim 2, as follows.

Ground 1 (Naveh renders Element 1[e1] obvious):

104. Under the first ground, the power-performance tuning table is the “performance state transition table” in Figure 3 of Naveh. Ex-1005 ¶11.

34

Combined Optimal State				Shared Feature SpeedStep® Control				Independent Feature TM1 Control		
Pindex	Perf	Abs-Pwr	Pwr%	Frequency	Vcc	Pwr%	Perf	Throttle%	Pwr%	Perf
0	1	100	100%	1700	1.233	100%	100%	0	1	100%
1	0.97	84	84%	1600	1.183	84%	97%	0	1	100%
2	0.94	68	68%	1500	1.133	68%	94%	0	1	100%
3	0.91	54	54%	1400	1.083	54%	91%	0	1	100%
4	0.88	41	41%	1300	1.008	41%	88%	0	1	100%
5	0.85	28	28%	1200	0.933	28%	85%	0	1	100%
6	0.82	16	16%	1100	0.85	16%	82%	0	1	100%
7	0.79	15	15%	1000	0.85	15%	79%	0	1	100%
8	0.76	13	13%	900	0.85	13%	76%	0	1	100%
9	0.73	12	12%	800	0.85	12%	73%	0	1	100%
10	0.7	10	10%	700	0.85	10%	70%	0	1	100%
11	0.67	9	9%	600	0.85	9%	67%	0	1	100%
12	0.58625	8	8%	600	0.85	9%	67%	0.125	0.9063	88%
13	0.5025	7	7%	600	0.85	9%	67%	0.25	0.8125	75%
14	0.41875	6	6%	600	0.85	9%	67%	0.375	0.7188	63%
15	0.335	6	6%	600	0.85	9%	67%	0.5	0.625	50%
16	0.1675	4	4%	600	0.85	9%	67%	0.75	0.4375	25%

50 52 42 44 36 38 40

FIG. 3

105. As described above for Element 1[e2], coordination logic 16 receives operating requirements 18 and 20 and selects one as the shared resource setting 22. *See supra* Section VII.A.2.f.ii. Naveh teaches that coordination logic 16 uses operating requirements 18 and 20 (one of which is shared resource setting 22) to index into the table. Ex-1005 ¶23 (“[O]perating requirements 18, 20 could also identify a performance requirement such as a performance level or an index into a performance state transition table.”).

106. Naveh gives an example of using the 100% performance to index into the table and extract the target frequency/voltage value of 1700 Mhz/1.233V: “The coordination logic can therefore use the 100% performance to index into the combined optimal state portion of the table, extract the target frequency/voltage value of 1700 MHz/1.233V from the table, and place it on the control bus 66.” *Id.*

¶32. The 100% performance corresponds to “1” in the second column 52 of the annotated Figure 3 above. That entry corresponds to the 1700 MHz and 1.233V in columns 36 and 38.

107. Naveh further describes using the information from the table to update at least one setting:

If the illustrated system is in the “max detect” mode (i.e., Force_Min is not set), the first shared resource control block 64 compares the two operating requirements and determines that the appropriate global setting is 100% performance. Accordingly, the target frequency/voltage value of 1700 MHz/1.233V may be selected as the shared resource setting 68. In the illustrated example, the shared resource setting 68 is applied to the clock source and/or voltage source, instructing the shared resources to operate at the appropriate frequency and/or core voltage defined by the shared resource setting 68.

Id. ¶33.

108. The index into the table is shared resource setting 22, which is the global EPB value as explained above for Element 1[e2] (Section VII.A.2.f.ii). Because shared resource setting 22 is selected based, in part, on the workload configuration input (as explained above for Element 1[e1] (Section VII.A.2.f.i), coordination logic 16 accesses the table based on both the workload configuration input and the global EPB value as recited in claim 2.

Ground 2 (Naveh in view of Saeed renders claim 2 obvious):

109. The above Ground 1 argument also applies to Ground 2. Moreover,

Saeed provides additional disclosures for dependent claim 2, further supporting that the Naveh-Saeed combination renders claim 2 obvious.

110. Saeed teaches that workload information (such as the level of multi-threadedness) may be used to control the performance state of a multicore processor. Ex-1006, 2:9-11 (“For one embodiment of the invention, the workload may be defined by the level of multi-threadedness.”), *id.*, 3:13-16 (“The power control circuit 210 receives as input a measured workload from the operating system. The workload may be defined by the level of multi-threadedness of the IC 200.”). Thus, this workload information (such as the level of multi-threadedness) is a workload configuration input.

111. Saeed further teaches that power control circuit 210 may use the workload configuration input (such as the level of multi-threadedness) to index a “lookup table” to identify a “corresponding” frequency or voltage setting for the cores 220, 230, and 240:

The frequency setting may be a function of the level of multi-threadedness measured in operation 410. Depending on the level of multi-threadedness of the system, *a lookup table* or a logic circuit may be used to determine a corresponding frequency that is between the minimum frequency and the maximum frequency. Similarly, voltages of the CPU cores may be adjusted based on the level of multi-threadedness measured.

Id., 4:8–15 (emphasis added).

112. It would have been obvious to a POSITA to apply the teachings of Saeed to Naveh. *See supra* Section VII.A.1. Both Saeed and Naveh describe using a table to determine the frequency and voltage setting of the cores. Ex-1005, Fig. 3, ¶¶23,32; Ex-1006, 4:8–15. Naveh teaches indexing the table using a global EPB (shared resource setting 22). Ex-1005 ¶¶23, 32. Saeed teaches indexing the table using a workload configuration input (e.g., the level of multi-threadedness). Ex-1006, 4:8–15. Based on these teachings, it would have been obvious to use a two-dimensional table that can be indexed independently by both a global EPB value and a workload configuration input. A POSITA would have been motivated to do so to improve the power management of the cores. By indexing the table based on two variables, the table would include settings based on two independent variables and thus the setting extracted from the table would correlate more closely to the desired operating conditions of the cores, resulting in improved performance and power savings. A POSITA would have recognized that Naveh and Saeed provide complimentary teachings, and could be combined without substantial changes in their respective functions because they teach substantially similar systems. *See supra* Section VII.A.1.

113. Thus, the Naveh-Saeed combination renders obvious dependent claim 2.

- b. Claim 3: The processor of claim 2, wherein the tuning circuit is to update the at least one setting of the at least one power management feature responsive to the global EPB value.**

114. Naveh teaches dependent claim 3 because claim 3 is encompassed by the limitations of claim 1. *See supra* Section VII.A.2.

- c. Claim 19: The processor of claim 1, wherein the power controller is to enable a first power management feature based on the updated at least one setting.**

115. Naveh teaches dependent claim 19. As described above with respect to Element 1[e3], Naveh teaches “updat[ing] at least one setting of at least one of the plurality of power management features.” *See supra* Section VII.A.2.f.iii. Naveh also discloses “enabl[ing] a first power management feature based on the updated at least one setting.” Naveh states that “the target values for the resource settings can be identified by *feature*.” Ex-1005 ¶31 (emphasis added). For example, with reference to Figure 3, Naveh states, “the Speedstep® feature (‘Shared Feature’) may provide control over shared resource settings, which are illustrated as clock frequency settings (‘Frequency’) 36 and core voltage settings (‘Vcc’) 38.” *Id.* ¶31. Thus, Naveh teaches that coordination logic 16 enables a first power management feature (the Speedstep® feature) based on the updated at least one setting (the frequency and voltage settings). *See supra* Sections VII.A.2.f.iii and VII.A.3.a. To the extent that “enabl[ing] a first power management feature” is

interpreted to mean turning on or off the feature, Naveh also teaches turning on or off a feature. For example, in Figure 3 of Naveh (illustrated below), a feature called “Throttle %” is set to zero (off) or non-zero (on).

34 ▼

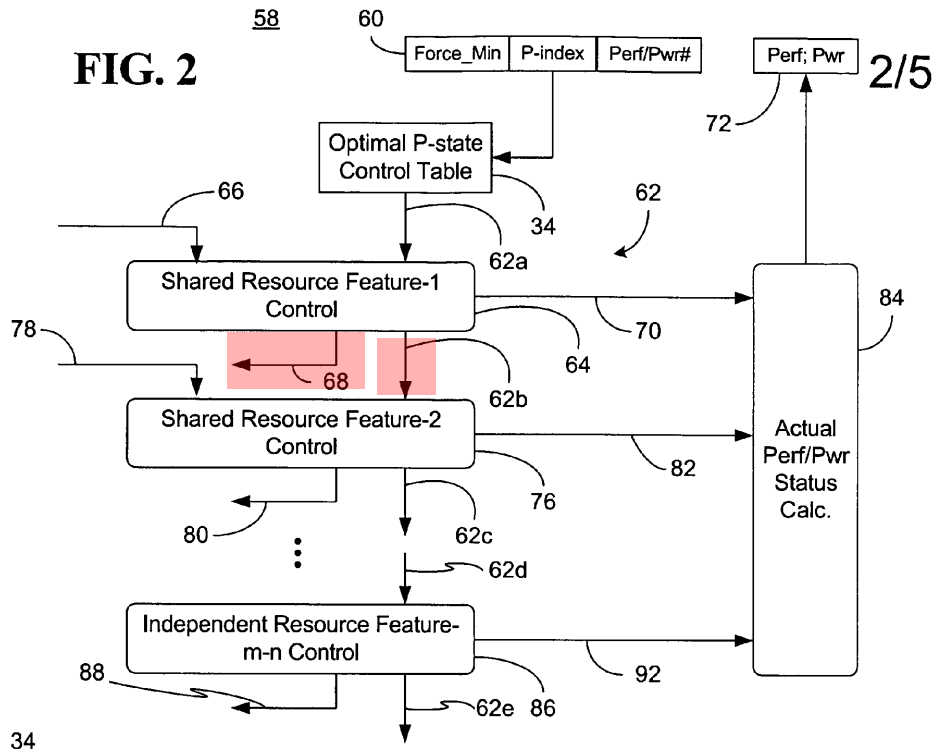
Combined Optimal State				Shared Feature SpeedStep® Control				Independent Feature TM1 Control		
Pindex	Perf	Abs-Pwr	Pwr%	Frequency	Vcc	Pwr%	Perf	Throttle%	Pwr%	Perf
0	1	100	100%	1700	1.233	100%	100%	0	1	100%
1	0.97	84	84%	1600	1.183	84%	97%	0	1	100%
2	0.94	68	68%	1500	1.133	68%	94%	0	1	100%
3	0.91	54	54%	1400	1.083	54%	91%	0	1	100%
4	0.88	41	41%	1300	1.008	41%	88%	0	1	100%
5	0.85	28	28%	1200	0.933	28%	85%	0	1	100%
6	0.82	16	16%	1100	0.85	16%	82%	0	1	100%
7	0.79	15	15%	1000	0.85	15%	79%	0	1	100%
8	0.76	13	13%	900	0.85	13%	76%	0	1	100%
9	0.73	12	12%	800	0.85	12%	73%	0	1	100%
10	0.7	10	10%	700	0.85	10%	70%	0	1	100%
11	0.67	9	9%	600	0.85	9%	67%	0	1	100%
12	0.58625	8	8%	600	0.85	9%	67%	0.125	0.9063	88%
13	0.5025	7	7%	600	0.85	9%	67%	0.25	0.8125	75%
14	0.41875	6	6%	600	0.85	9%	67%	0.375	0.7188	63%
15	0.335	6	6%	600	0.85	9%	67%	0.5	0.625	50%
16	0.1675	4	4%	600	0.85	9%	67%	0.75	0.4375	25%

50 52 42 44 36 38 40

FIG. 3

- d. **Claim 20:** The processor of claim 1, wherein the power controller is to send an update message to a destination agent of the processor to cause the destination agent to control a first power management feature based on the updated at least one setting.

116. Naveh teaches dependent Claim 20. For example, Figure 2 shows “a power management architecture 58 that can be used to implement hardware coordination logic” 16. Ex-1005 ¶29.



Naveh states that “the shared resource setting 68 is applied to the clock source and/or voltage source, instructing the shared resources to operate at the appropriate frequency and/or core voltage defined by the shared resource setting 68.” *Id.* ¶33. Based on this disclosure, Naveh teaches that a power controller (coordination logic 16) sends an update message to a destination agent (clock source and/or voltage source) of the processor to cause the destination agent (clock source and/or voltage source) to control a first power management feature (core frequency or voltage) based on the updated at least one setting (shared resource setting).

117. Alternatively, Naveh also states:

The first shared resource control block 64 can generate corrected operating requirements and/or target values

based on the shared resource setting 68 and place the corrected target values on the control bus 62b, where the corrected target values enable a second shared resource control block 76 to determine a shared resource setting 80 for the next shared resource. ... In the illustrated example, the shared resource setting 80 could be a bus frequency setting.

Id. ¶34. Based on this disclosure, Naveh teaches that a power controller (coordination logic 16) sends an update message to a destination agent (second shared resource control block 76) of the processor to cause the destination agent (second shared resource control block 76) to control a first power management feature (bus frequency) based on the updated at least one setting (corrected target values based on the shared resource setting 68).

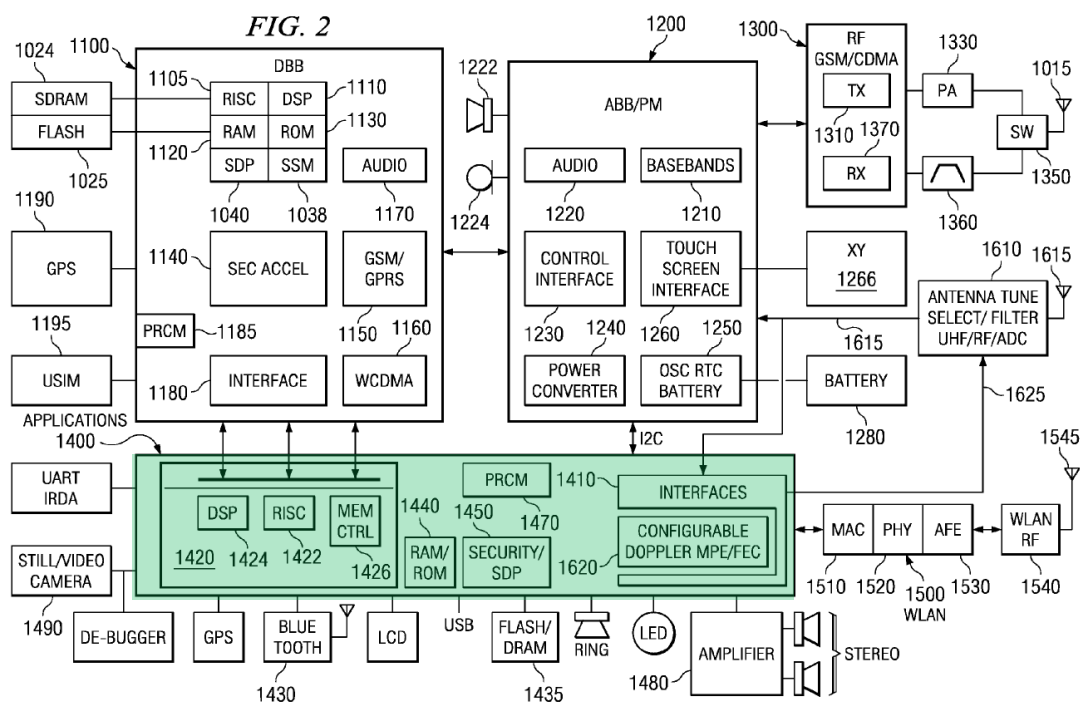
B. Ground 3: Claims 1–6, 11, 13, and 19–20 Are Obvious Over Titiano

1. Independent Claim 1

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

118. To the extent that the preamble of claim 1 is limiting, Titiano discloses the preamble. Titiano discloses a processor, which is usually identified as applications processor integrated circuit 1400 in the various figures. Ex-1007, Figs. 2, 3, 4, 5, 8, and 9. For example, annotated Figure 2 shows “an integrated circuit chip or core 1400 ... for applications processing and more off-chip peripherals.”

Id. ¶63.

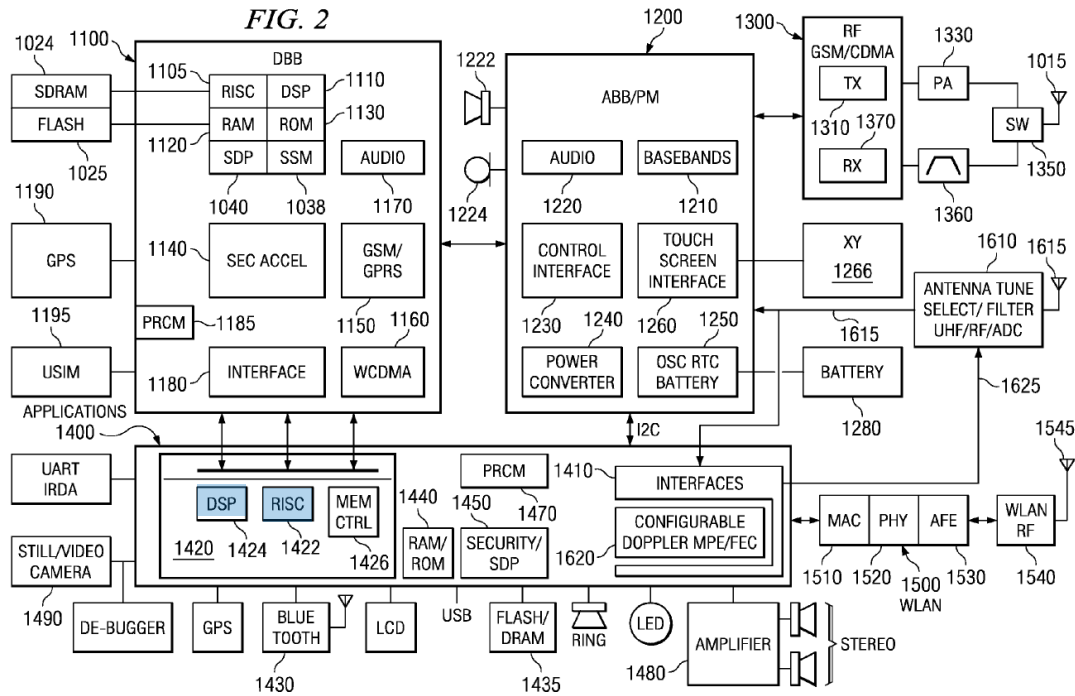


Titiano describes Figure 3 as illustrating “a block diagram of an inventive applications processor integrated circuit in FIG. 2 with associated integrated circuits.” *Id.* ¶20.

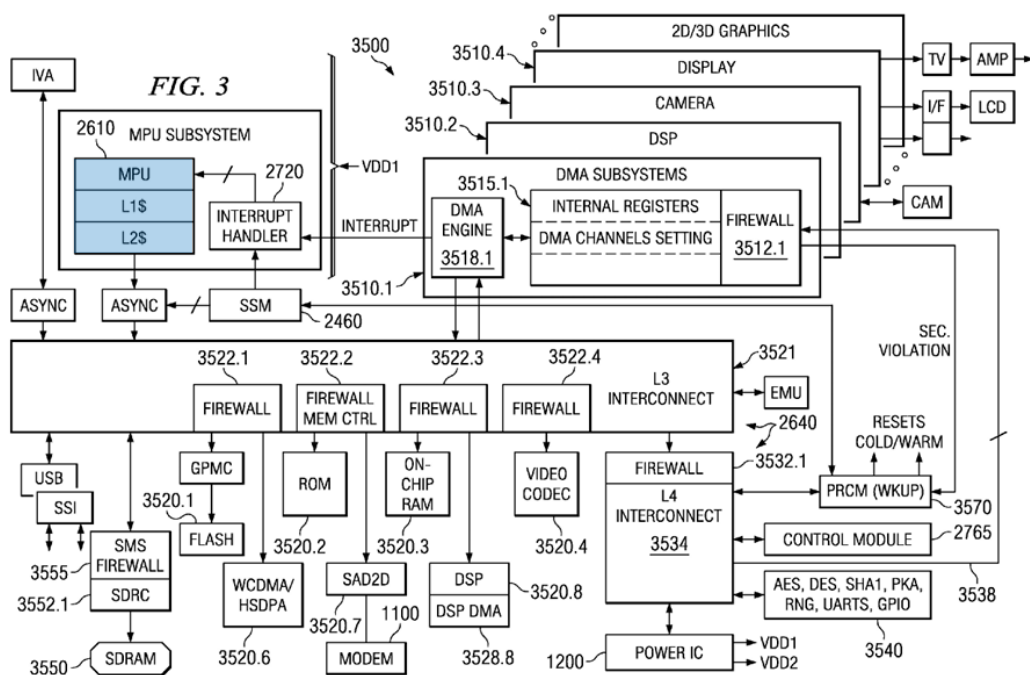
b. Element 1[a]: a plurality of cores;

119. Titiano discloses Element 1[a]. The applications processor integrated circuit chip in Titiano includes a plurality of cores. For example, with respect to Figure 2, Titiano states that “provided on chip 1400 is an applications processing section 1420 which includes a RISC processor 1422 (such as MIPS core(s), ARM core(s), or other suitable processor), [and] a digital signal processor (DSP) 1424 such as from the TMS320C55x™ DSP generation and/or the TMS320C6x™ DSP generation from Texas Instruments Incorporated or other digital signal

processor(s).” Ex-1007 ¶63.



120. Similarly, the processor IC in Figure 3 includes MPU subsystem, which “has one or more *processors with CPUs* such as RISC or CISC processors 2610, and having superscalar processor pipeline(s) with L1 and L2 caches.” *Id.* ¶76 (emphasis added). The CPUs in processors 2610 are a plurality of cores, as shown in annotated Figure 3.



121. For the “power controller” claim elements addressed below, this Declaration relies on the power management system in Figures 8–15 and 18–19, which Titiano states is “for use” in the systems of Figures 2 and 3. *Id.* ¶¶25–32, 35–36. Thus, the power management system in Figures 8–15 and 18–19 applies to the cores in Figures 2 and 3. Titiano illustrates those cores as boxes labeled “CPU” in Figures 8, 9, 10, 13 and 15 and as boxes labeled “CORE” in Figures 18 and 19. Annotated Figures 8, 10, and 13 below provide examples.

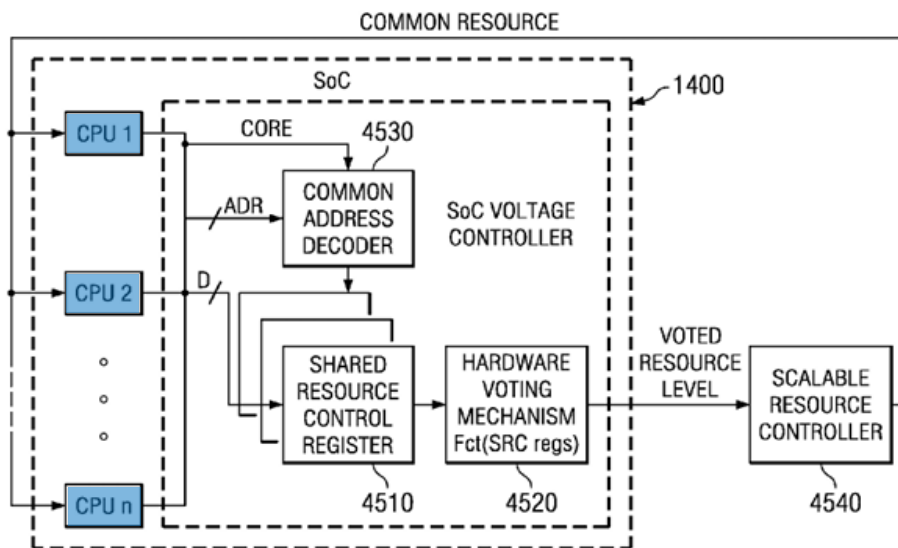


FIG. 8

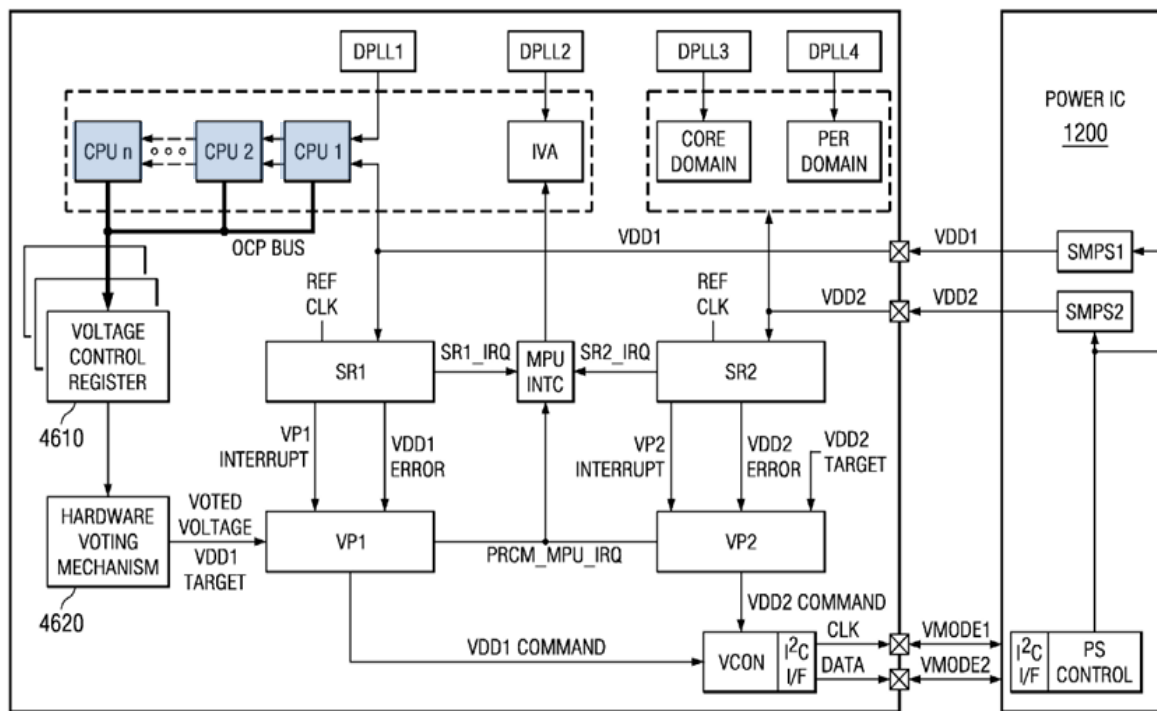


FIG. 10

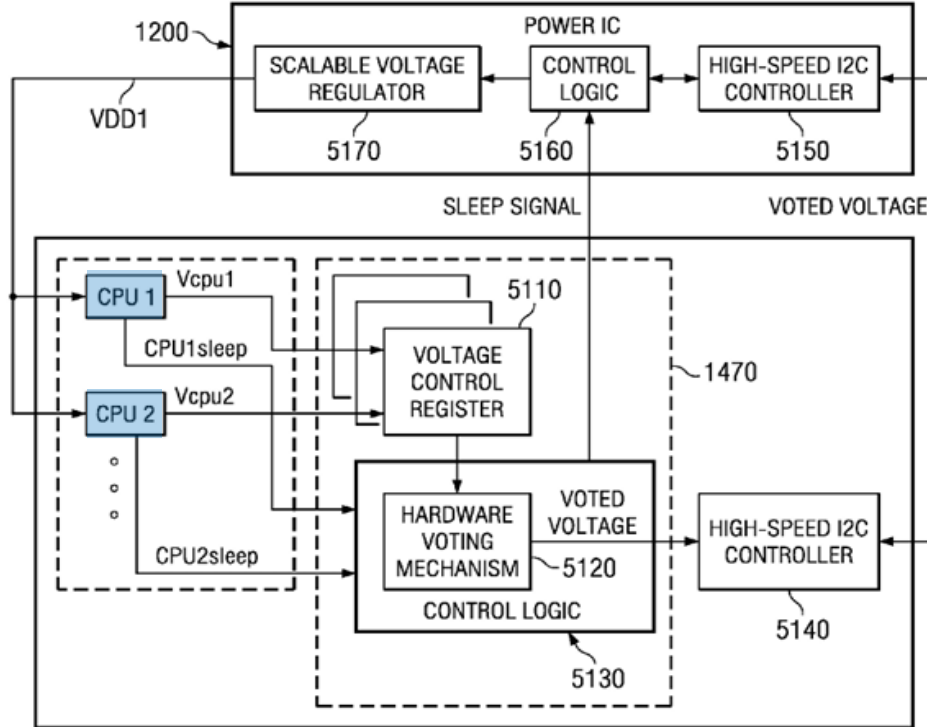
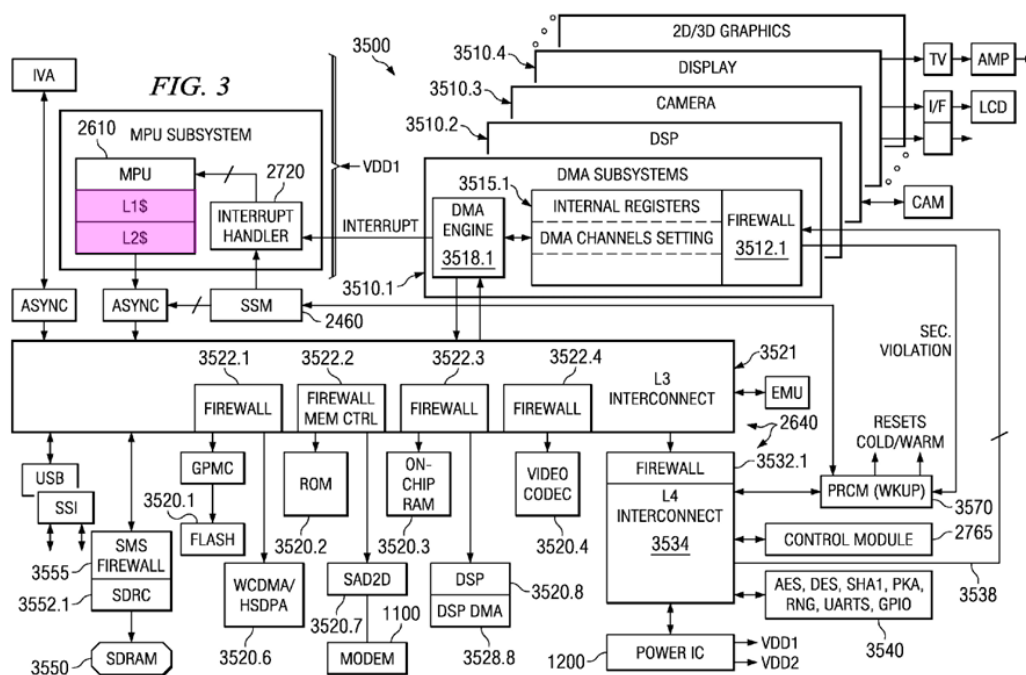


FIG. 13

122. Thus, Titiano discloses “a plurality of cores,” as recited in Element 1[a].

c. Element 1[b]: a cache memory;

123. Titiano discloses Element 1[b]. Titiano teaches that the RISC or CISC processors 2610 in Figure 3 have “superscalar processor pipeline(s) with *L1 and L2 caches*.” Ex-1007 ¶76 (emphasis added).



124. Thus, Titiano discloses “a cache memory,” as recited in Element 1[b].

d. Element 1[c]: an interconnect to couple the plurality of cores and the cache memory;

125. Titiano teaches Element 1[c]. As described above, Figure 3 of Titiano discloses a processor with a plurality of cores and L1 and L2 caches. Ex-1007 ¶76. The cores must be able to access the cache memory, which requires an interconnect. As discussed above, a POSITA would understand Titiano's L2 cache to be shared by all the cores of the processor. *See supra* Section VI.A. Thus, for the multiple cores to access and use the L2 cache, there must necessarily be an interconnect coupling the cores to the L2 cache. Accordingly, it is inherent that the processor would also include an interconnect to couple the cores and cache

memory.

126. Alternatively, it would have been obvious to modify Titiano to include an interconnect to couple the plurality of cores to the L2 cache memory. For example, Titiano discloses that the cores “are operable for inter-processor communication, such as by an OCP [Open Core Protocol] bus.” Ex-1007 ¶155. Therefore, a POSITA would have recognized that an OCP bus could be used as an interconnect to couple the cores together and to a cache memory.

e. Element 1[d]: a power controller to control a plurality of power management features of the processor;

127. Titiano discloses Element 1[d]. In Titiano, Figures 8–15 and 18–19 illustrate varying levels of detail and, thus, the claimed power controller corresponds to different blocks with different names in the figures. For example, with reference to Figures 5, 10, 12, 13, and 19, Titiano describes the power controller as Power Reset and Clock Manager (PRCM) 1470. I note that although Figure 10 does not include a reference number or label for PRCM, Titiano describes Figure 10 as including PRCM. Ex-1007 ¶¶163, 165. Also, Titiano refers to PRCM 1470 in Figure 13 as the “Power Management Control Module 1470.” *Id.* ¶¶185, 186. With reference to Figures 8 and 9, Titiano illustrates the power controller as the SoC Voltage Controller.

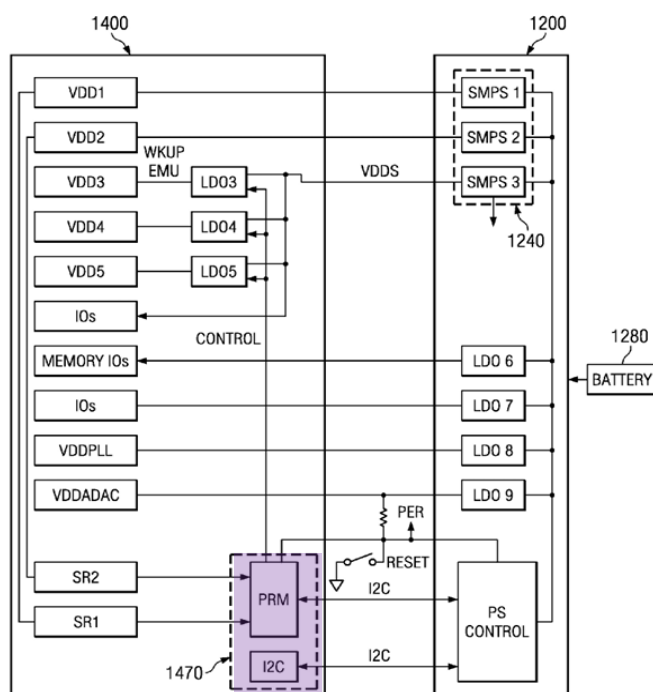


FIG. 5

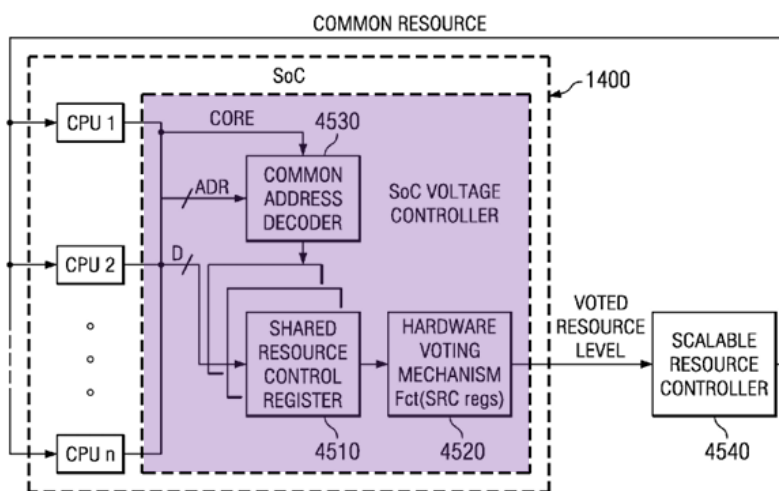
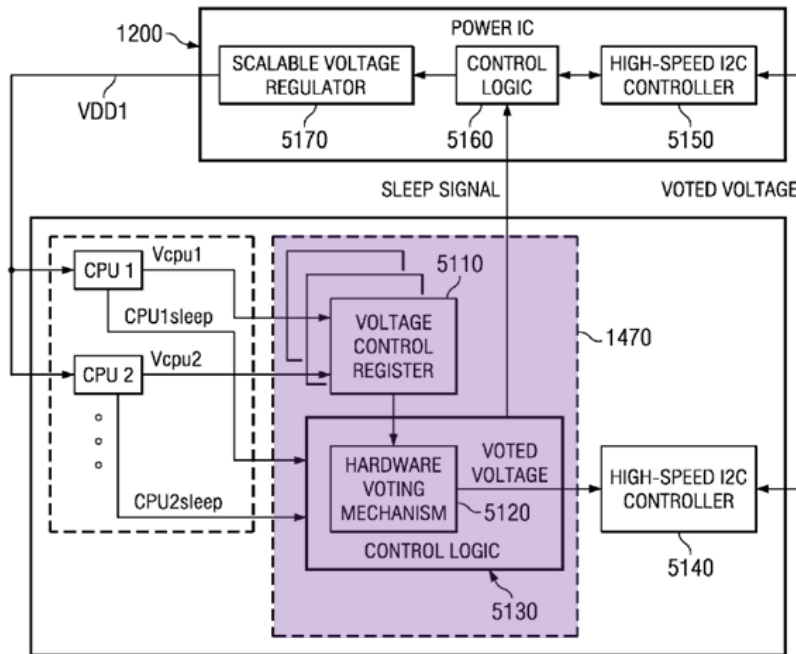
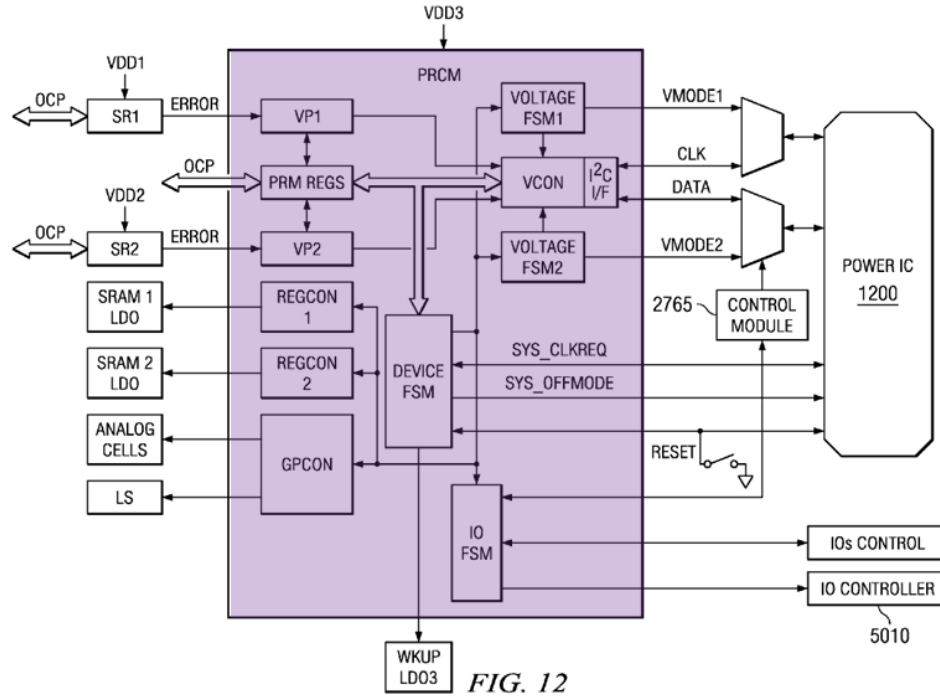


FIG. 8



128. Regardless of the nomenclature, the claimed “power controller” corresponds to the portion of the applications processor integrated circuit chip that

controls the power delivered to the processor. *See, e.g., id.* ¶¶119–120. The power controller is used to control a plurality of power management features such as frequency and voltage, as described in further detail below with reference to Element 1[e]. *See infra* Section VII.B.1.f. That analysis focuses on PRCM 1470 in Figures 10, 12, 13, and 19.

- f. Element 1[e]: wherein the power controller includes a tuning circuit to [e1] receive a workload configuration input regarding a workload, [e2] receive a plurality of energy performance bias (EPB) values and determine a global EPB value based thereon, and [e3] update at least one setting of at least one of the plurality of power management features based on the workload configuration input and the global EPB value.**

129. Titiano teaches Element 1[e]. In Figures 10, 12, 13, and 19 of Titiano, the power controller is PRCM 1470. The claimed “tuning circuit” is the portion of PRCM 1470 that performs the functions recited in Elements 1[e1], 1[e2] and 1[e3].

- In Figure 10, the tuning circuit includes voltage control registers 4610, hardware voting mechanism 4620, and voltage processor VP1.
- In Figures 12 and 19, the tuning circuit includes voltage control registers 4610 (which are included among the PRM registers), voltage processor VP1, and voltage controller VCON.
- In Figure 13, the tuning circuit includes voltage control registers 5110 and control logic 5130 with hardware voting mechanism 5120.

As described below, the tuning circuit in Titiano performs the functions recited in Elements 1[e1], 1[e2] and 1[e3].

i. Element 1[e1]: a tuning circuit to receive a workload configuration input regarding a workload;

130. Titiano teaches Element 1[e1]. This Declaration present two mappings for the “workload configuration input” in Elements 1[e1] and 1[e3]. One mapping is based on Titiano’s power management feature called Dynamic Power Switching (DPS); the other mapping is based on Titiano’s power management feature called Adaptive Voltage Scaling (AVS). Ex-1007 ¶106. Both mappings are presented below.

Dynamic Power Switching (DPS)

131. Titiano describes DPS as a power management feature that switches the cores to a sleep or low-power standby mode when they are not active. *Id.* ¶114. To implement DPS, the tuning circuit in Titiano receives a signal indicating whether the cores have an active workload, and that signal is a “workload configuration input regarding a workload.”

132. Titiano states that “[b]ased on the hardware resources activity, as indicated by FIFO buffer status, for instance, DPS switches the device dynamically from any of several medium power modes to a lower power mode during application execution or to a low power Standby mode.” *Id.* ¶114; *see also id.*

(“DPS provides automatic hardware detection of conditions (e.g. FIFO fill status or IO hardware signals) for sleep and wakeup transition for clock and power.”). DPS also “predicts system load and switches the module clocks dynamically between On and Off modes.” *Id.* A POSITA would have further understood that DPS is used not only to assess the current workload of the CPU cores, but also to predict their future workload. For example, Titiano describes that DPS “software predicts dynamically the performance requirement of the application(s) running on the processor,” and “accounts for the wakeup latencies overhead due to domain switching so that DPS is operative when such latencies do not significantly impact the performance of the device.” *Id.* ¶211. From this disclosure, a POSITA would have understood that the DPS software takes into account whether the CPU cores are expected to be busy in the near future, and in which case it would be inefficient to place them into a sleep state only to wake them up shortly thereafter.

133. As illustrated in annotated Figure 13, control logic 5130 receives from each CPU core a signal indicating whether the CPU has an active workload (CPU1sleep, CPU2sleep). *Id.* ¶188.

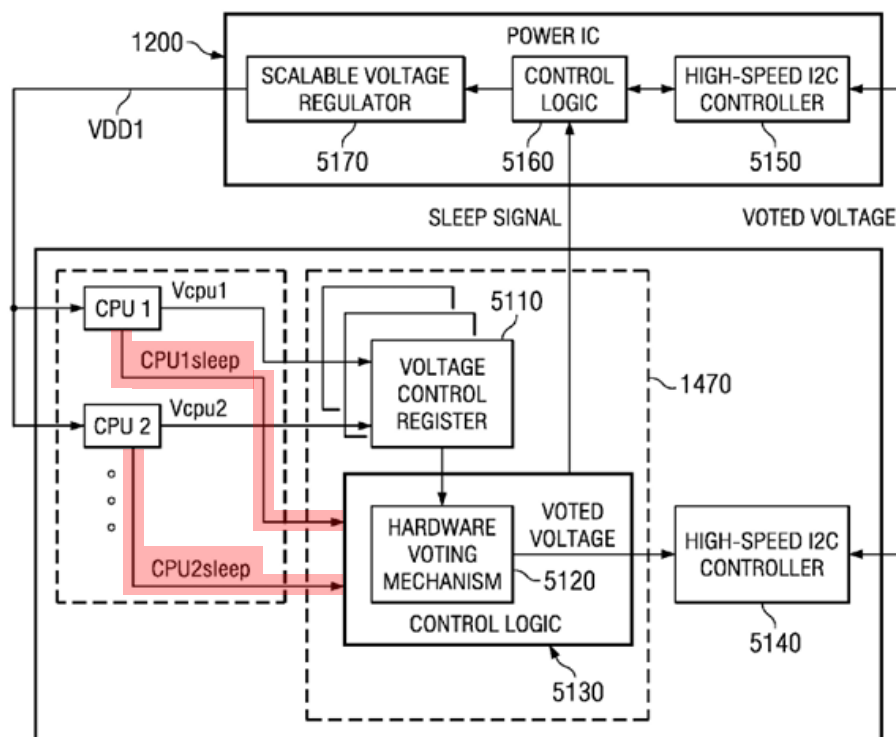


FIG. 13

For example, if the FIFO buffer status of CPU 1 is not empty, CPU1sleep signal indicates that CPU 1 has an active workload, and if the FIFO buffer status of CPU 1 is empty, CPU1sleep signal indicates that CPU 1 does not have an active workload. *Id.* ¶¶114, 208–209, 217. Thus, the tuning circuit of Titiano (control logic 5130) receives a workload configuration input (CPU1sleep or CPU2sleep signal) regarding a workload.

134. As described in further detail below with reference to Element 1[e3], the tuning circuit of Titiano (control logic 5130) uses the workload configuration input to determine when all the cores are inactive so that the cores' power domain including those cores can be switched to the sleep voltage. *Id.* ¶185 (“When the

Sleep Signal is active (all CPUs asleep), the Power IC sets the voltage to Sleep voltage. When the Sleep signal is inactive (at least one of the CPUs awake), the Power IC sets the voltage to the Voted Voltage.”); *id.* ¶191 (“The arbitrating circuit generates the sleep output designated SLEEP SIGNAL and feeds it to control logic 5160 of FIG. 13 for controlling the shared resource (e.g., of supply voltage) when all the processors in the set of potentially-voting processors are asleep in the applicable domain.”).

135. Thus, Titiano’s implementation of DPS teaches Element 1[e1].

Adaptive Voltage Scaling (AVS)

136. Titiano describes AVS as a power management feature that adjusts the power supply voltage based, in part, on “the temperature induced current performance of the device.” *Id.* ¶174. To implement AVS, the tuning circuit in Titiano receives a signal indicating “the temperature induced current performance of the device” and that signal is a “workload configuration input.”

137. Titiano states that “Adaptive power supply AVS reduces active power consumption.” *Id.* ¶134. Using AVS, “[t]he power supply voltage is adapted to the silicon performance either statically (depending on the device manufacturing process), or dynamically (depending on the temperature induced current performance of the device). An AVS sub-chip uses sensors to monitor the silicon performance and outputs a measure of the performance error.” *Id.* The operation of

AVS is illustrate below in annotated Figure 10.

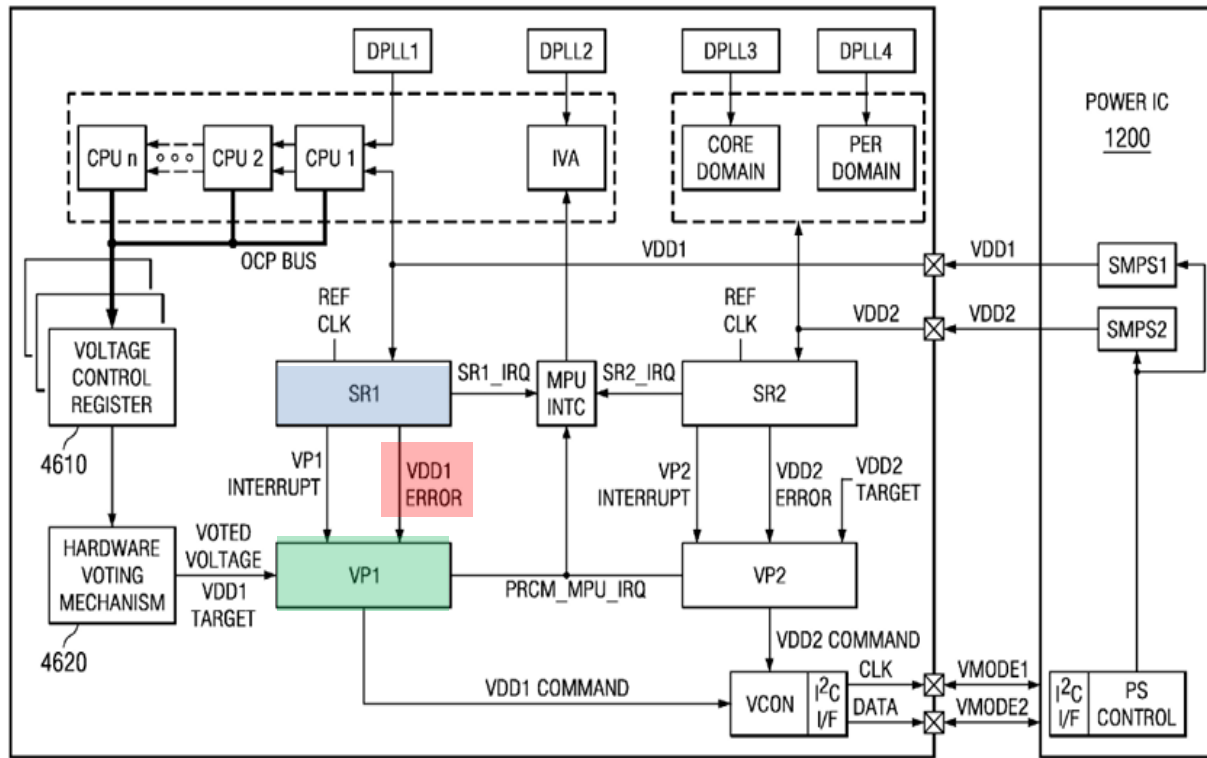


FIG. 10

As shown in Figure 10, CPU 1 and CPU 2 reside in a single voltage domain that both receive the same voltage VDD1. *Id.* ¶153, Fig. 10. Sensor Module SR1 generates a performance error signal VDD1 Error that it sends to the Voltage Processor VP1. *Id.* ¶¶177–78.

138. The VDD1 Error signal indicates “the temperature induced current performance of the device” (*Id.* ¶134), and thus is a “workload configuration input regarding a workload.” A POSITA would understand that temperature is an indication of the load on and activity of the processor. This is because higher

processor activity causes more transistors to switch (i.e. turn on and off) more frequently. Power consumption is based on the frequency and number of transistors that are switching and an increase in power consumption means an increase in temperature. Thus, non-CPU intensive activities, such as reading email (where the CPU is often idle and causing fewer transistors to be switching) results in lower temperatures. Similarly, CPU intensive activities, such as video streaming or gaming (where the CPU is never idle and more transistors are switching), results in higher temperatures. Using AVS, temperature sensors (e.g., the sensor modules of Titiano) monitor the temperature of the processor as a whole or the individual CPU cores. This allows the processor to adjust to the thermal conditions that the processor is currently experiencing. The VDD1 Error signal is a representation of temperature (which indicates workload), allowing the processor to adjust voltage as a function of temperature. Thus, the tuning circuit of Titiano (Voltage Processor VP1) receives a workload configuration input regarding a workload (VDD1 Error).

139. To the extent that the “workload configuration input” must be based on a user input, it would have been obvious to modify Titiano to allow the user to control a workload configuration input. For example, Titiano describes that “[w]hen performing DVFS, the user can program a new operating point in the sensor module by programming the new frequency target and caus[ing] the sensor module to re-calculate the error.” *Id.* ¶135. Then, “[t]he Voltage Processor takes

the average frequency error from the AVS sub-chip and determines the appropriate voltage level to program into the power supply.” *Id.* Therefore, it would have been obvious to a POSITA that the user can input the AVS performance error by setting a new operating point, as described in Titiano.

140. As described in further detail below with reference to Element 1[e3], the tuning circuit of Titiano (Voltage Processor VP1) uses the workload configuration input to adjust the setting (VDD1 Command) provided to the voltage controller VCON. *Id.* ¶¶135, 156.

141. Thus, Titiano’s implementation of AVS teaches Element 1[e1].

ii. Element 1[e2]: a tuning circuit to ... receive a plurality of energy performance bias (EPB) values and determine a global EPB value based thereon;

142. Titiano teaches Element 1[e2]. Titiano describes another power management feature, called Dynamic Voltage and Frequency Scaling (DVFS), which dynamically adjusts the voltage and frequency of the cores to reduce power consumption. Ex-1007 ¶¶107, 109. The tuning circuit in Titiano receives voltage requests from the cores and picks the highest of the voltage requests as the “voted voltage,” which it uses to update the voltage setting for the cores. The voltage requests received from the cores are “a plurality of energy performance bias (EPB) values” and the voted voltage is the “global EPB value.”

143. Titiano states that the CPU cores submit requests for a voltage level sufficient to satisfy their operating requirements. *Id.* ¶12. The CPUs write their voltage requests in registers (which are called shared resource control registers 4510 in Figure 8 and voltage control registers 4610/5110 in Figures 9, 10, 12, 13, 15 and 19). *Id.* ¶¶136, 137, 138, 146, 165, 196. Note that Figures 12 and 19 illustrate PRM registers, but Titiano states that these registers “include Voltage Control Registers 4610 ... and are used ... for DVFS.” *Id.* ¶¶165 (for Fig. 12), 220 (stating Fig. 19’s structures are analogous to Fig. 12). These registers are shaded red in the annotated figures below.

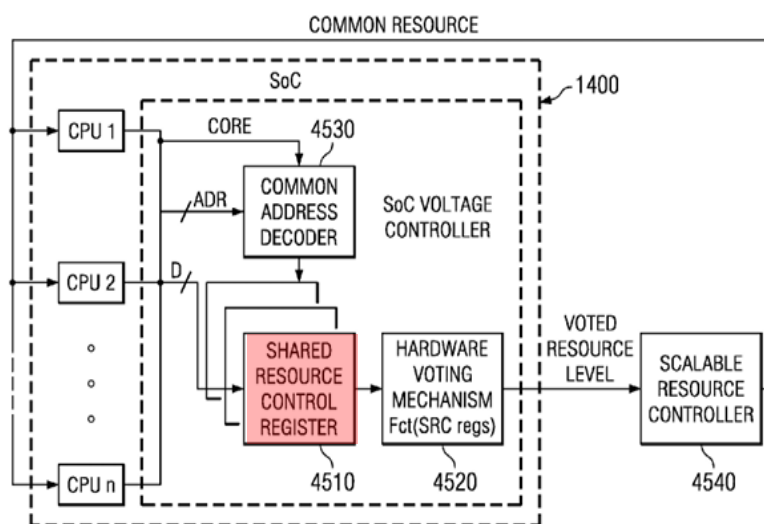
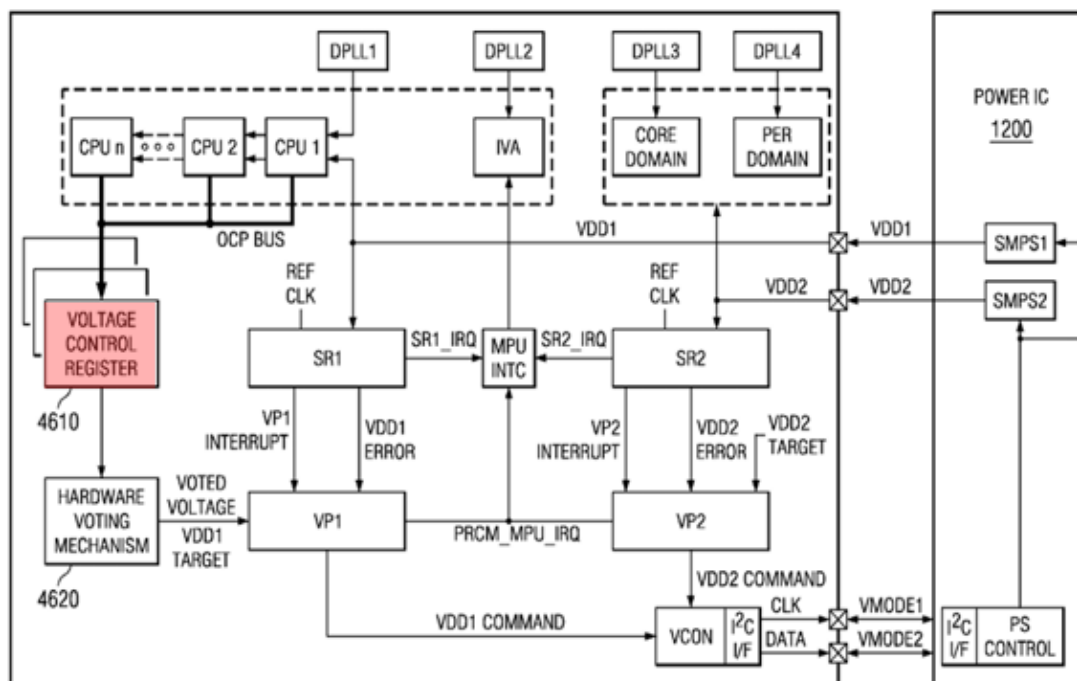
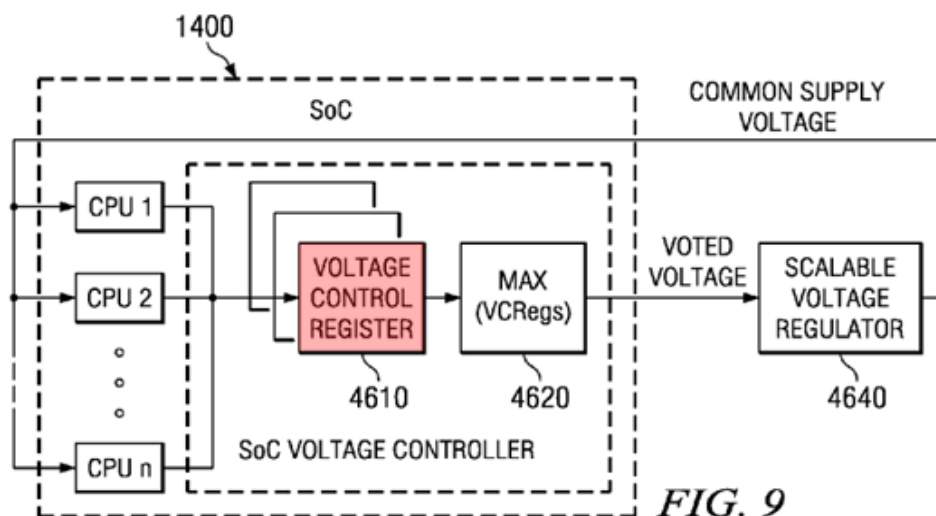


FIG. 8



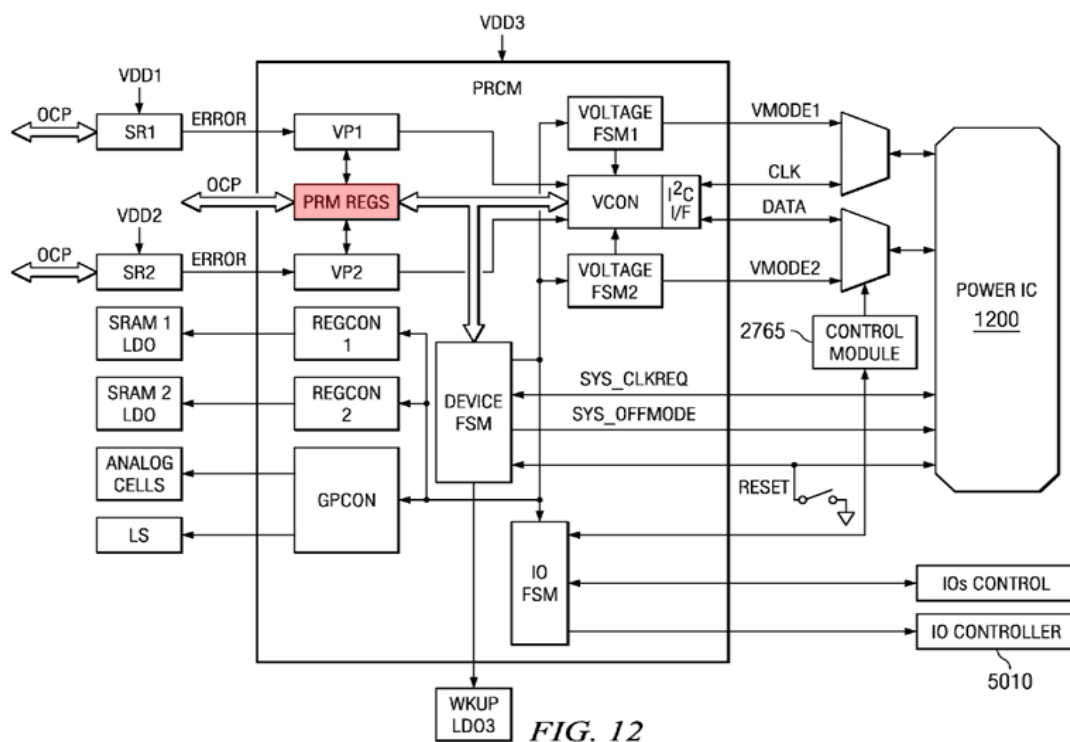


FIG. 12

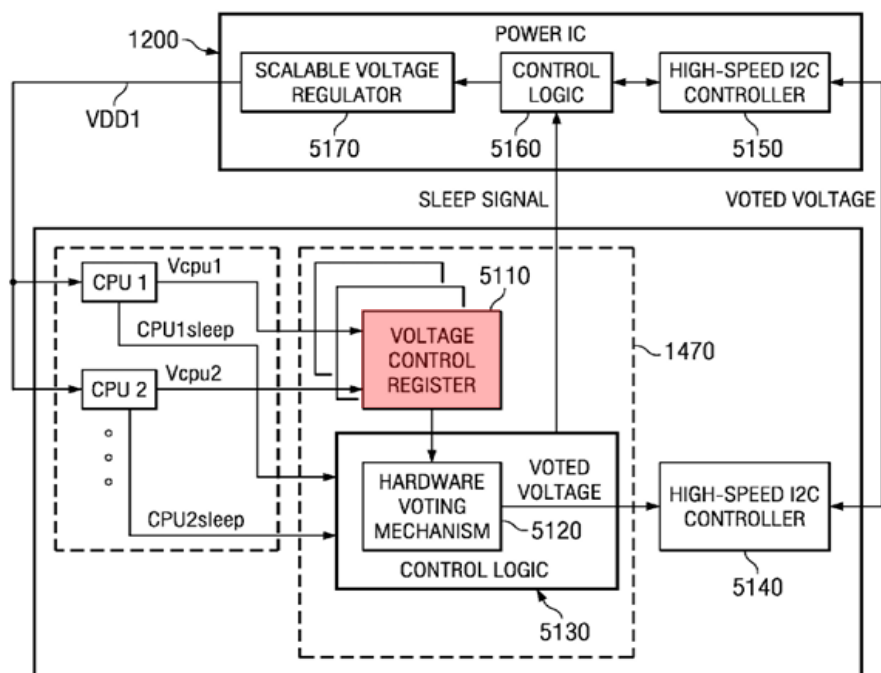


FIG. 13

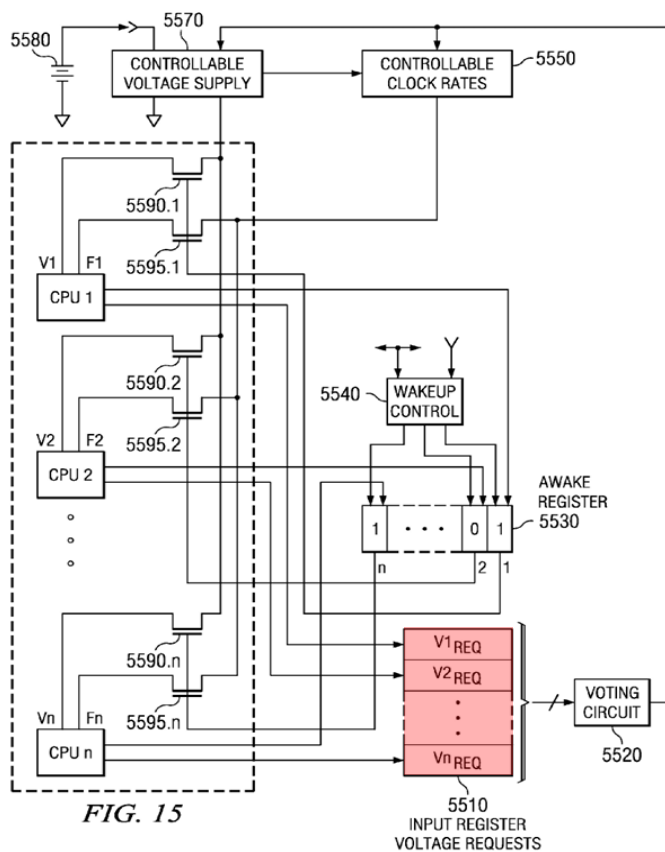


FIG. 15

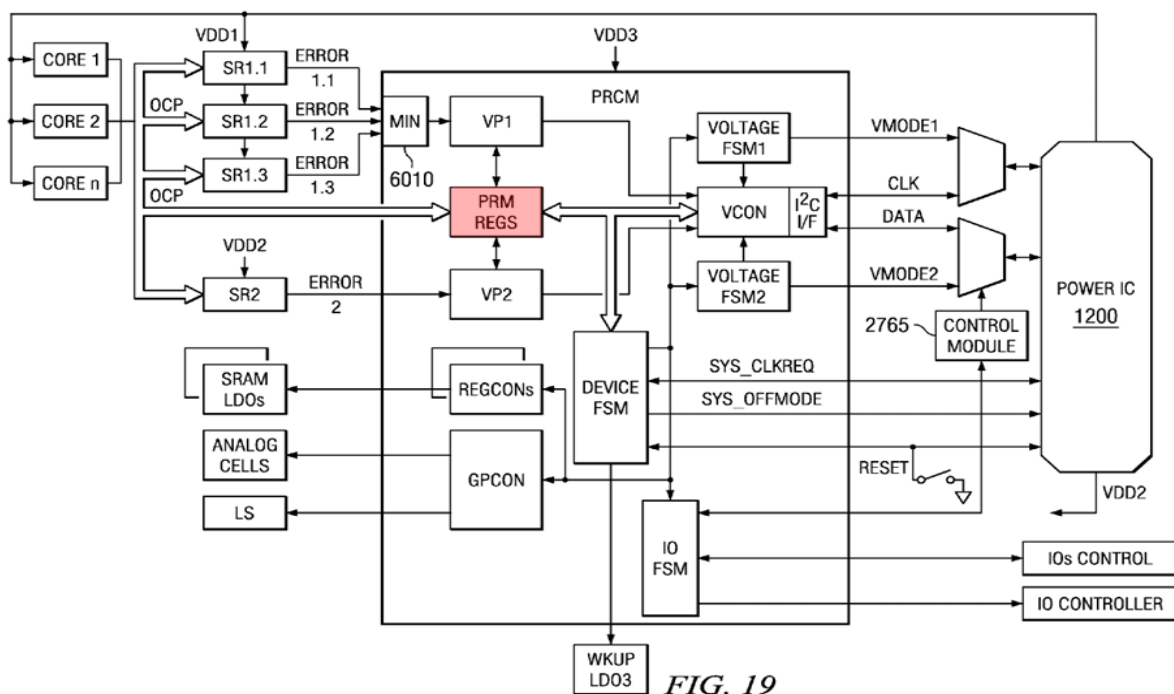


FIG. 19

These registers are part of the tuning circuit of Titiano. Thus, just as the '197 Patent states that tuning circuit 100 may “receive incoming EPB values from multiple cores of a multicore processor” (Ex-1005 4:59–60), the tuning circuit in Titiano receives a plurality of EPB values (requested voltages) from the cores.

144. In Titiano, arbitrating circuits select the maximum of the voltage requests as the voted voltage. Ex-1007 ¶136 (“The hardware voting mechanism 4520 selects the applicable one of the power management requirements or requests thus written[.]”); *id.* ¶¶137–38 (“the highest requested voltage”), *id.* ¶146 (“the highest requested voltage”); *id.* ¶150 (“the maximum of the values in the registers in the Voltage Control Registers block 4610”); *id.* ¶¶170, 199 (“voting circuit 5520 call[s] for the maximum or highest voltage requested by any processor in the voltage domain”). The arbitrating circuits are identified by green shading the figures below. The arbitrating circuit is hardware voting mechanism 4520 in Figure 8; hardware voting mechanism 4620 in Figures 9 and 10; hardware voting mechanism 5120 in Fig. 13; and hardware voting circuit 5520 in Figure 15. In Figures 12 and 19, the arbitrating circuit for VDD1 is voltage FSM1. *Id.* ¶¶ 165, 175 (stating that Fig. 12 implements DVFS); *id.* ¶173 (stating that voltage FSM 1 manages VDD1); *id.* ¶220 (stating Fig. 19’s structures are analogous to Fig. 12).

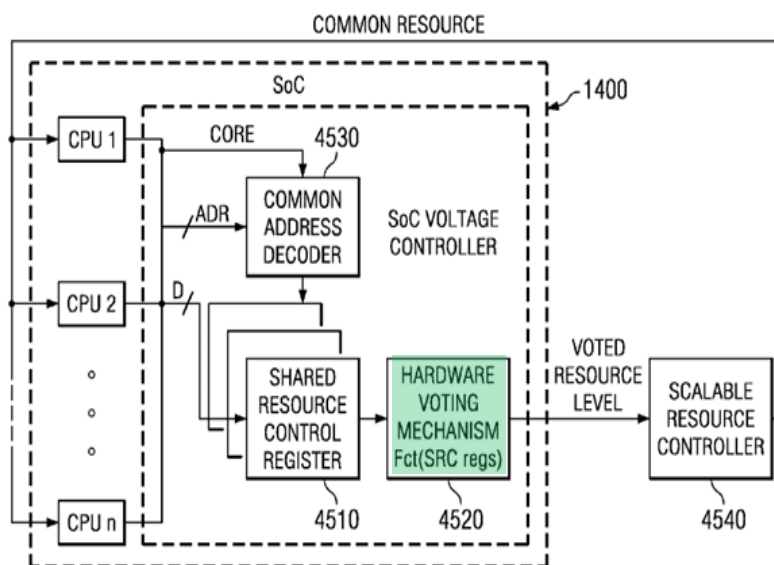


FIG. 8

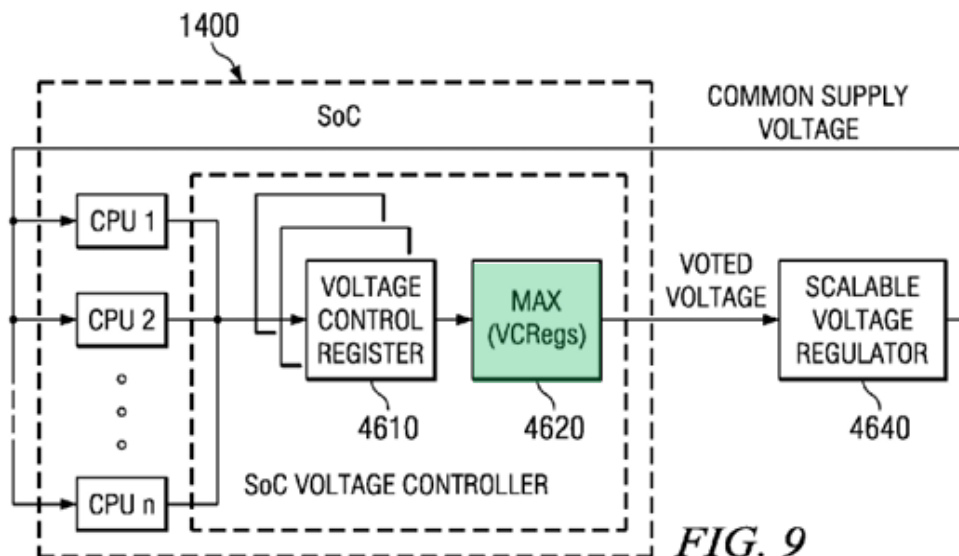


FIG. 9

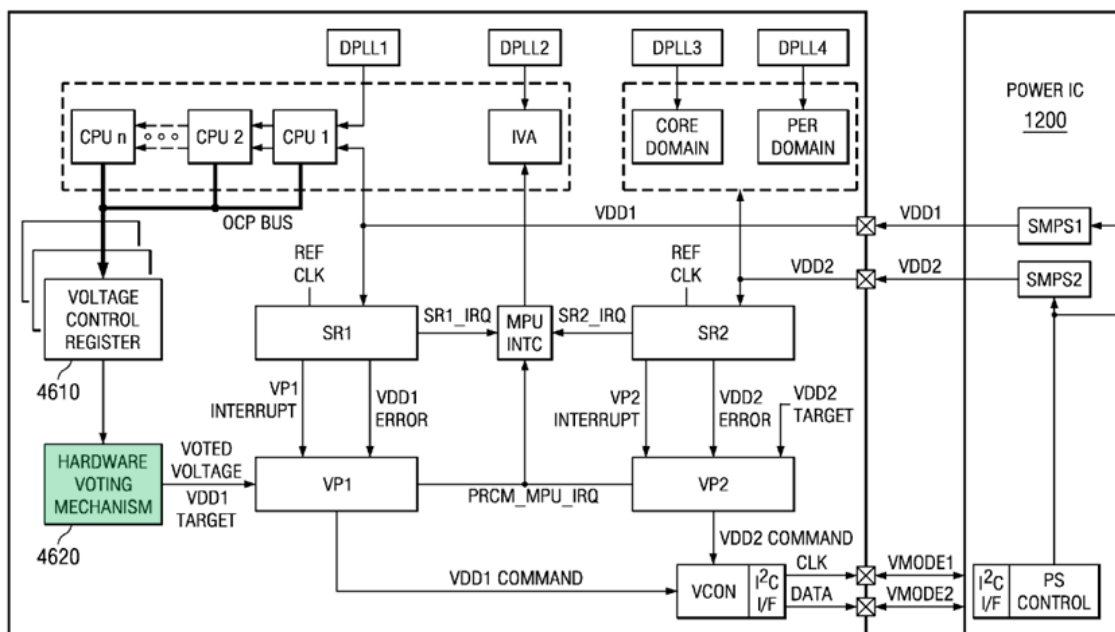


FIG. 10

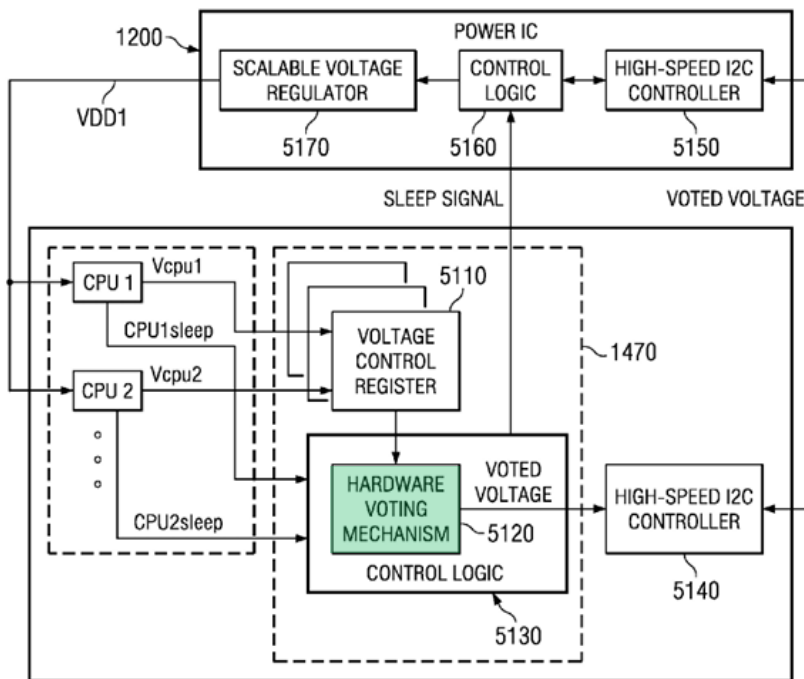
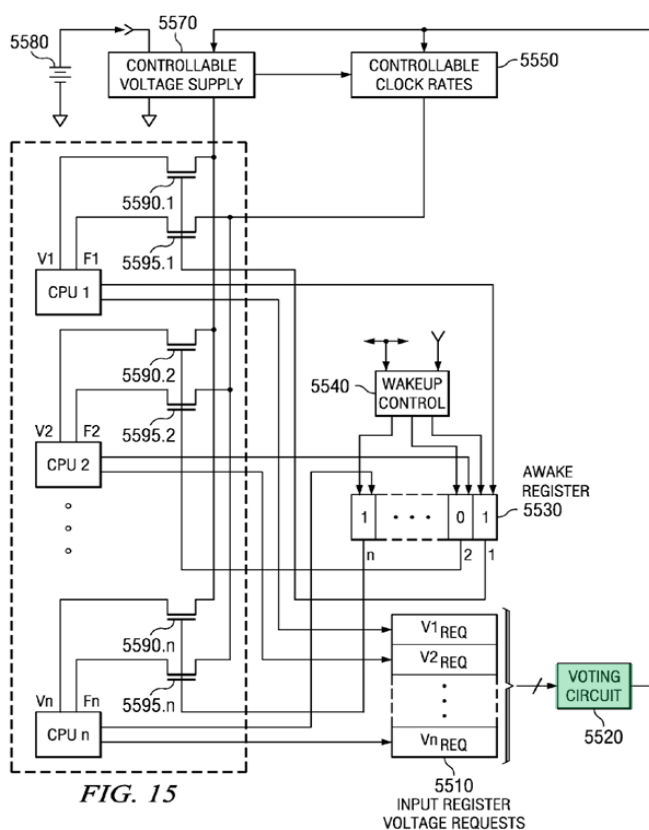


FIG. 13



These arbitrating circuits are part of the tuning circuit of Titiano. Thus, the tuning circuit of Titiano determines a global EPB value (voted voltage) based on the plurality of EPB values (the voltage requests from the cores).

145. According to the '197 Patent, energy performance bias (EPB) values represent a tradeoff between power and performance. Ex-1001, 2:34–38, 3:49–51, 4:58–64. In Titiano, the voltage requests and the highest voted voltage represent a tradeoff between power and performance and, thus, are EPB values. Ex-1007 ¶¶122 (“When medium application performance is desired or when application performance requirements vary, the DVFS technique is applied. The voltage and

frequency are scaled to match the closest OPP that meets the performance requirement based on a voting process.”).

146. The highest voted voltage selected by the arbitrating circuit is a “global” EPB value because, as described with reference to Element 1[e3], the tuning circuit of Titiano uses the voted voltage to update the global voltage setting for the cores. Ex-1005 ¶137 (“The electronic circuitry further has a scalable resource controller 4540 responsive to the arbitrating circuit, e.g. 4520, to deliver the shared resource to the set of processors CPU 1, CPU2, . . . CPU_n at substantially the arbitrated resource level of the shared resource represented by the multi-bit value output from the arbitrating circuit, e.g. 4520.”); *id.* ¶138 (“Hardware voting mechanism 4520 then automatically delivers a command or Voted Resource Level to scale the voltage as specified by that function (e.g., the highest requested voltage as in FIG. 9 or minimum error as in FIG. 19) or includes the circuitry 4540 itself to scale the voltage to the highest request voltage.”); *id.* ¶186 (“the CPU cores are provided in and as a shared voltage domain”); *id.* ¶146 (“The hardware voting mechanism 4620 then automatically scales the voltage to the highest requested voltage.”).

147. Accordingly, Titiano discloses Element 1[e2].

iii. Element 1[e3]: a tuning circuit to ... update at least one setting of at least one of the plurality of power management features based on the workload configuration input and the global EPB value.

148. Titiano teaches Element 1[e3]. This Declaration presents two mappings for Element 1[e3]. One is based on Titiano's DPS and DVFS, and the other is based on Titiano's AVS and DVFS. Titiano teaches that the power controllers may implement DPS and DVFS or AVS and DVFS. Ex-1007 ¶11 ("Some implementations combine any two or more of DVFS, DPS and AVS power management processes in order to optimize the power saving in all modes of operation and for all sources of power consumption."); *id.* ¶106 ("Combined DVFS and DPS are established in power management structure and process. A power-efficient DVFS/DPS result is achieved with minimum voltage, hence low power.").

DPS and DVFS

149. Titiano teaches that DPS and DVFS may be combined to update a setting of a power management feature (VDD1 Common Supply Voltage) based on the workload configuration input (CPU1sleep or CPU2sleep signal) and the global EPB value (max voted voltage).

150. As shown in annotated Figure 13, control logic 5130 sends two control signals—Sleep Signal and Voted Voltage—to the Power IC 1200.

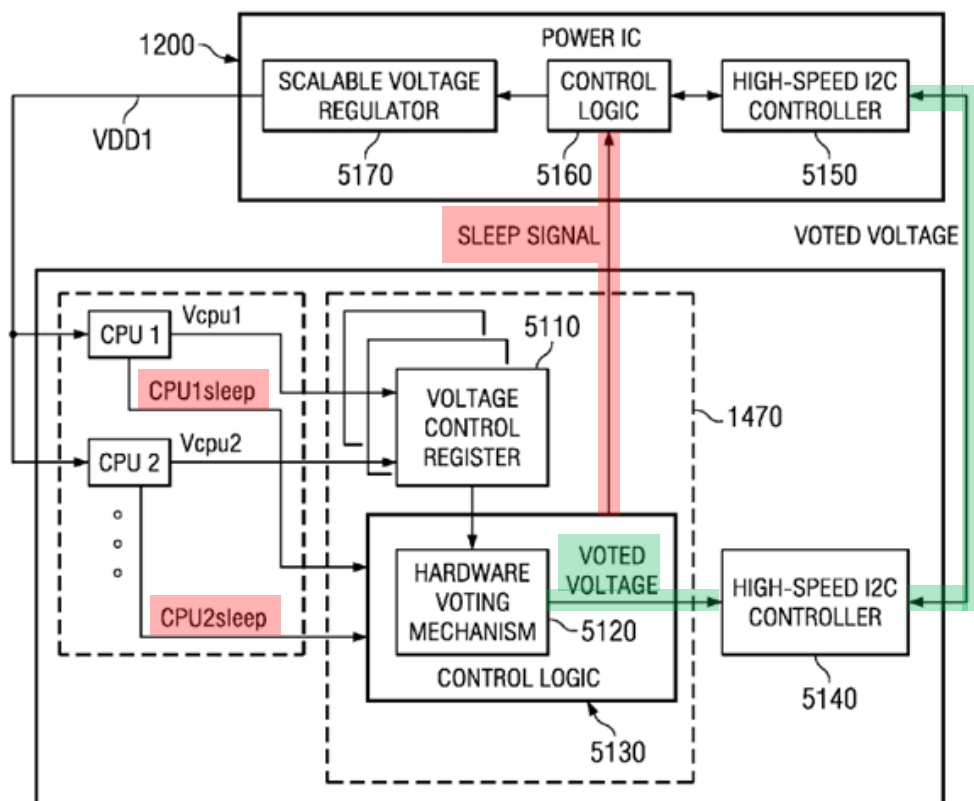


FIG. 13

The Sleep Signal is based on CPU1sleep or CPU2sleep signals received from the CPU cores. Ex-1007 ¶¶190–91.

151. Figure 14 illustrates the operation and timing of the control process implemented by control logic 5130 in Figure 13. *See id.* ¶¶190–92. As shown in the shaded green, the Max Voted Voltage is the maximum of the CPUs' voltage requests from the CPUs (V_{cpu1} and V_{cpu2}). *Id.* ¶191. As shown in the shaded red, the Sleep Signal is the minimum function of the CPU1sleep and CPU2sleep signals (meaning that the Sleep Signal is high only when both individual sleep signals are high). *Id.*

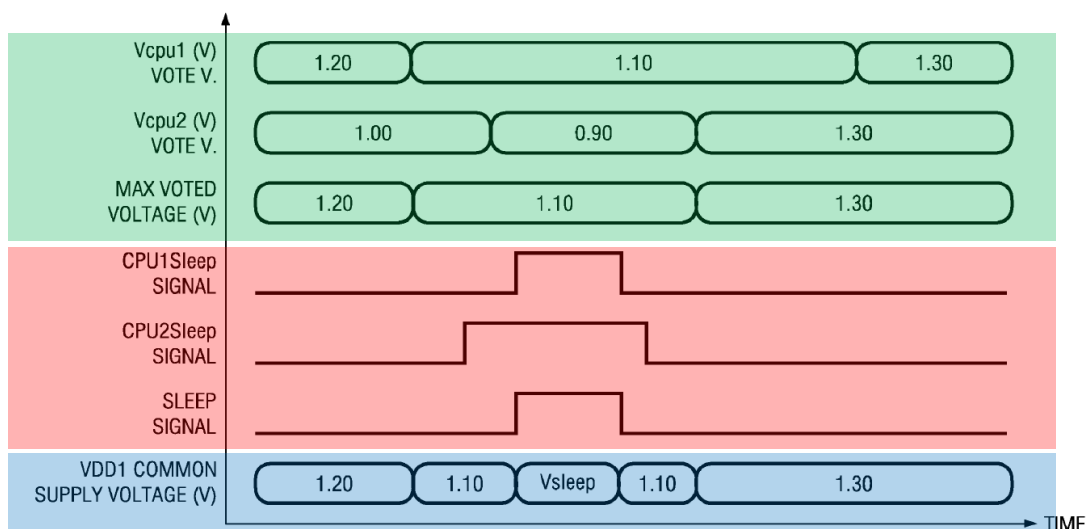


FIG. 14

152. Titiano describes the VDD1 Common Supply Voltage in shaded blue:

The bottom line in FIG. 14 is labeled VDD1 Common Supply Voltage, and represents the actual voltage delivered by the Scalable Voltage Regulator 5170 of FIG. 13 to the voltage domain including the CPU cores CPU1, CPU2, etc. This actual voltage corresponds to the Max Voted Voltage except that when SLEEP SIGNAL is active in FIG. 13, then an actual sleep voltage Vsleep is instead provided to all the CPU cores CPU1, CPU2, etc.

Id. ¶192. VDD1 is the claimed “at least one setting of at least one of the plurality of power management features.” As illustrated in Figure 13, VDD1 is based on CPU1sleep and CPU2sleep signals (the workload configuration input) and the Max Voted Voltage (the global EPB value). If CPU1sleep and CPU2sleep signals are both high (indicating the cores do not have an active workload), VDD1 is set to Vsleep. *Id.* ¶192. Otherwise, VDD1 is set to the Max Voted Voltage. *Id.* ¶191. Thus, the tuning circuit of Titiano (which includes control logic 5130) updates at

least one setting (VDD1) of at least one of the plurality of power management features based on the workload configuration input (CPU1sleep and CPU2sleep signals) and the global EPB value (Max Voted Voltage).

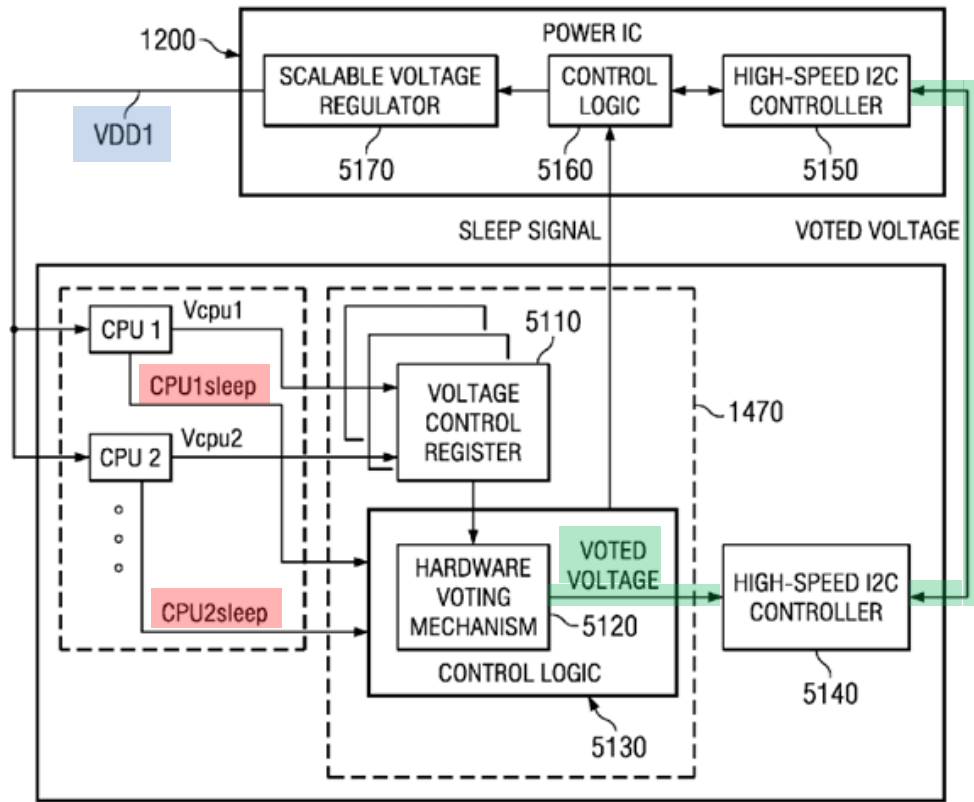
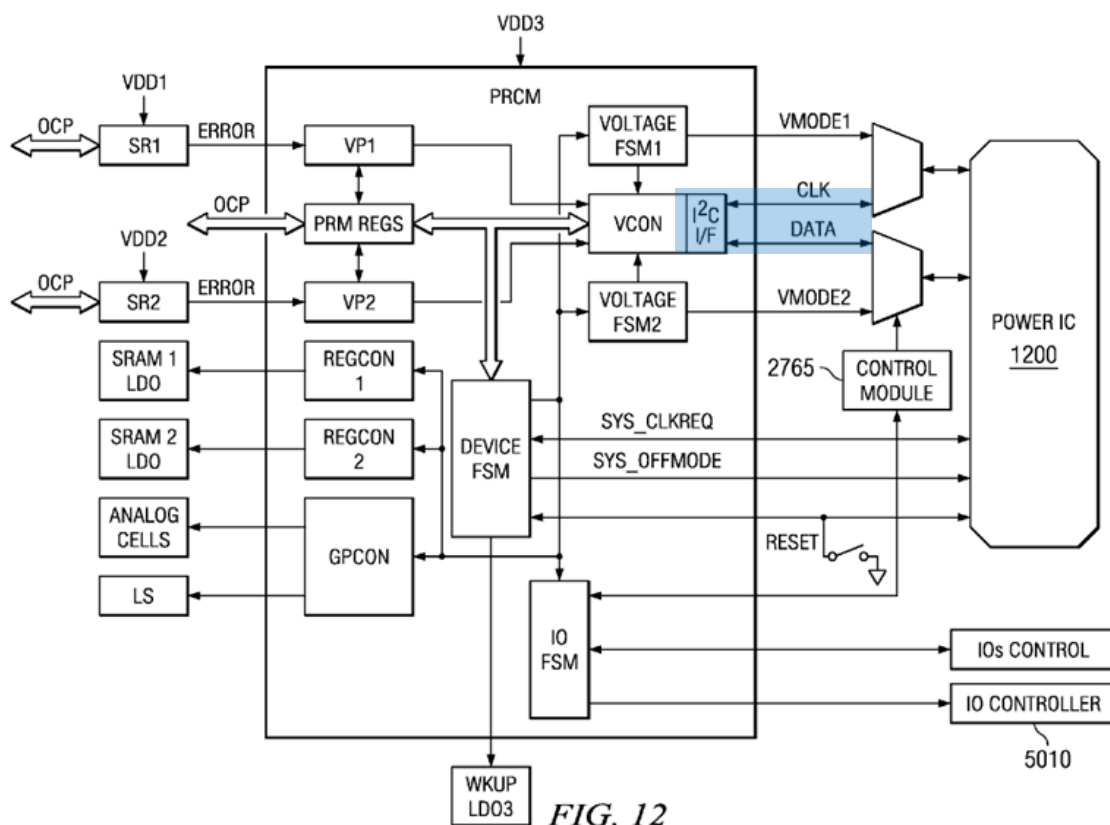


FIG. 13

153. Claim 1 does not preclude the tuning circuit from using another component (such as Power IC 120) to effectuate the claimed update. Control logic 5130 controls Power IC 1200 and thus is the component responsible for updating the setting VDD1. Power IC 1200 responds to instructions from control logic 5130 to set the value of VDD1.

154. To the extent that Claim 1 is read to require control logic 5130 to

effectuate the claimed update, Figure 12 of Titiano implements the functionality described in Figure 13 but uses a single control interface (the “I2C interface”) to control Power IC 1200, as illustrated below. Ex-1007 ¶154.



In Figure 12, the tuning circuit includes PRM control registers, voltage processors VP1, voltage FSM 1, and voltage controller VCON. This tuning circuit implements the DPS and DVFS functions described above: the voltage controller VCON sets the output voltage of Power IC 1200 based on whether the cores are active or inactive according to DPS and, if active, the voltage level according to DVFS (i.e., the voting mechanism). Titiano states that “[t]he VCON receives a voltage level

from the voltage processor and also a command from the voltage state machine FSM of FIG. 12. In active mode, the VCON formats the voltage level and command and sends them over the I2C interface to control the external [Switch Mode Power Supply] SMPS.” Ex-1007 ¶154. Conversely, “[w]hen the device is in standby mode, the voltage controller VCON is then driven by the voltage state machines FSMs to allow the external supply to enter low power mode.” *Id.* In I2C mode, the voltage controller VCON communicates the voltage setting to Power IC 1200 as Data over the I2C interface, and this communicated voltage setting is “at least one setting of at least one of the plurality of power management features based on the workload configuration input and the global EPB value,” as recited in Claim 1.

AVS and DVFS

155. Alternatively, Titiano’s combination of AVS and DVFS also teach Element 1[e3]. Titiano teaches that AVS and DVFS may be combined to update a setting of a power management feature (e.g., VDD1 Command in Figure 10) based on the workload configuration input (e.g., VDD1 Error in Figure 10) and the global EPB value (e.g., Voted Voltage in Figure 10). Ex-1007 ¶11. In Figure 10, the tuning circuit corresponds to the voltage control register 4610, hardware voting mechanism 4620 and voltage processor VP1.

156. Figure 10 illustrates the voltage voting process described in

connection with Figure 13, and the voted voltage is derived just as the “voted voltage” is in Figure 13. Figure 10 also illustrates sensor modules SR1 and SR2 used to implement AVS in the respective voltage domains of Voltage Processors VP1 and VP2. *Id.* ¶158.

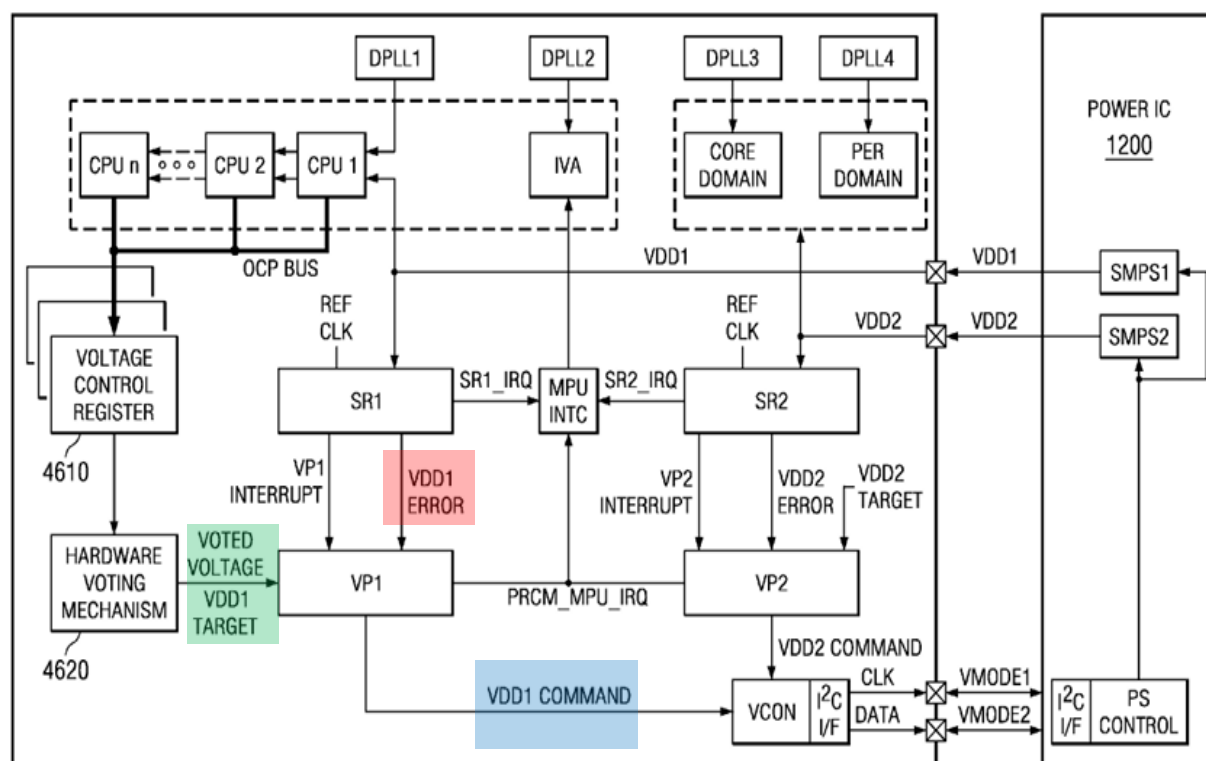


FIG. 10

SR1, for example, generates a VDD1 error and sends it to Voltage Processor VP1, which then uses the VDD1 error to “determine[] the appropriate voltage level to program into the power supply.” *Id.* ¶135; *see also id.* ¶155 (“In FIGS. 10 and 12, the digital circuit of sensor SR ... processes the difference between the N-count value and SR measured frequency and derives an error amount that indicates

whether to raise or lower the voltage.”). Voltage Processor VP1 receives the “error value from the AVS sensor module SR, and automatically adjusts the power supply by sending a voltage command to the Voltage Controller.” *Id.* ¶135. This voltage command is the “VDD1 Command” in Figure 10. *Id.*, Fig. 10.

157. Thus, Voltage Processor VP1 receives two signals: (1) the voted voltage from the hardware voting mechanism 4620; and (2) VDD1 Error from sensor module SR1. *Id.* Thus, Voltage Processor VP1 updates a setting (VDD1 Command) based on the workload configuration input (VDD1 Error) and the global EPB value (voted voltage). Accordingly, the VDD1 Command corresponds to “at least one setting of at least one of the plurality of power management features based on the workload configuration input and the global EPB value,” as recited in Claim 1.

2. Dependent Claims 2–6 and 19–20

- a. Claim 2: The processor of claim 1, wherein the power controller is to access a power-performance tuning table based on the workload configuration input and the global EPB value, and use information from the power-performance tuning table to update the at least one setting.**

158. Titiano teaches dependent claim 2. As described above, this Declaration presents two alternative mappings for the “workload configuration input”: one based on Titiano’s power management feature called Dynamic Power Switching (DPS) and another based on Titiano’s power management feature called

Adaptive Voltage Scaling (AVS). *See supra* Section VII.B.1.f. As described in detail below, these mappings both teach claim 2.

DPS and DVFS

159. Titiano discloses “a ***lookup table*** of OPP voltage-frequency pairs (V,F) to look up the frequency F corresponding to voltage V requested by active requesting core CPU_i and uses that frequency F from the lookup table as the operating frequency f_{CPU(i)} for CPU_i.” Ex-1007 ¶201 (emphasis added). This lookup table corresponds to the “power-performance tuning table” of claim 2.

160. As explained above for Element 1[e3], the voltage VDD1 is based on the workload configuration input (CPU1sleep and CPU2sleep signals) and the global EPB value (max voted voltage). *See supra* Section VII.B.1.f.iii. Titiano states that the voltage VDD1 corresponds to the max voted voltage, “except that when SLEEP SIGNAL is active in FIG. 13, then an actual sleep voltage V_{sleep} is instead provided to all the CPU cores CPU1, CPU2, etc.” *Id.* ¶192. Thus, if CPU1sleep and CPU2sleep signals are both high (indicating the cores do not have an active workload), VDD1 is set to V_{sleep}. *Id.* ¶192.

161. Titiano teaches “a lookup table of OPP voltage-frequency pairs (V,F) to lookup the frequency F corresponding to Voltage V.” *Id.* ¶201. It would have been obvious to index a table based on VDD1, which is based on the workload configuration input (CPU1sleep and CPU2sleep signals) and the global EPB value

(the max voted voltage). Thus, Titiano teaches that the power controller accesses the table based on the workload configuration input and the global EPB value.

162. The power controller in Titiano uses the information from the table to update a power management setting (the frequency associated with VDD1). *Id.*

¶109. Accordingly, Titiano's implementation of DPS and DVFS teaches claim 2.

AVS and DVFS

163. To implement AVS, Titiano describes that "[e]ach of the Voltage Processors contain Look-Up Tables and [are] correctly initialized and configured to optimize the voltage control." Ex-1007 ¶159. This look-up table corresponds to the "power-performance tuning table" of claim 2.

164. As explained above, AVS is used in conjunction with DVFS to generate "an error output that indicates whether to raise or lower the voltage." *Id.* ¶153; *see supra* Section VII.B.1.f. Titiano's Figure 10 illustrates an implementation of AVS. As explained above for Element 1[e3], the workload configuration input is the VDD1 error (which Voltage Processor VP1 receives from sensor SR1), and the global EPB is the voted voltage (which Voltage Processor VP1 receives from hardware voting mechanism 4620). *See supra* Section VII.B.1.f.iii.

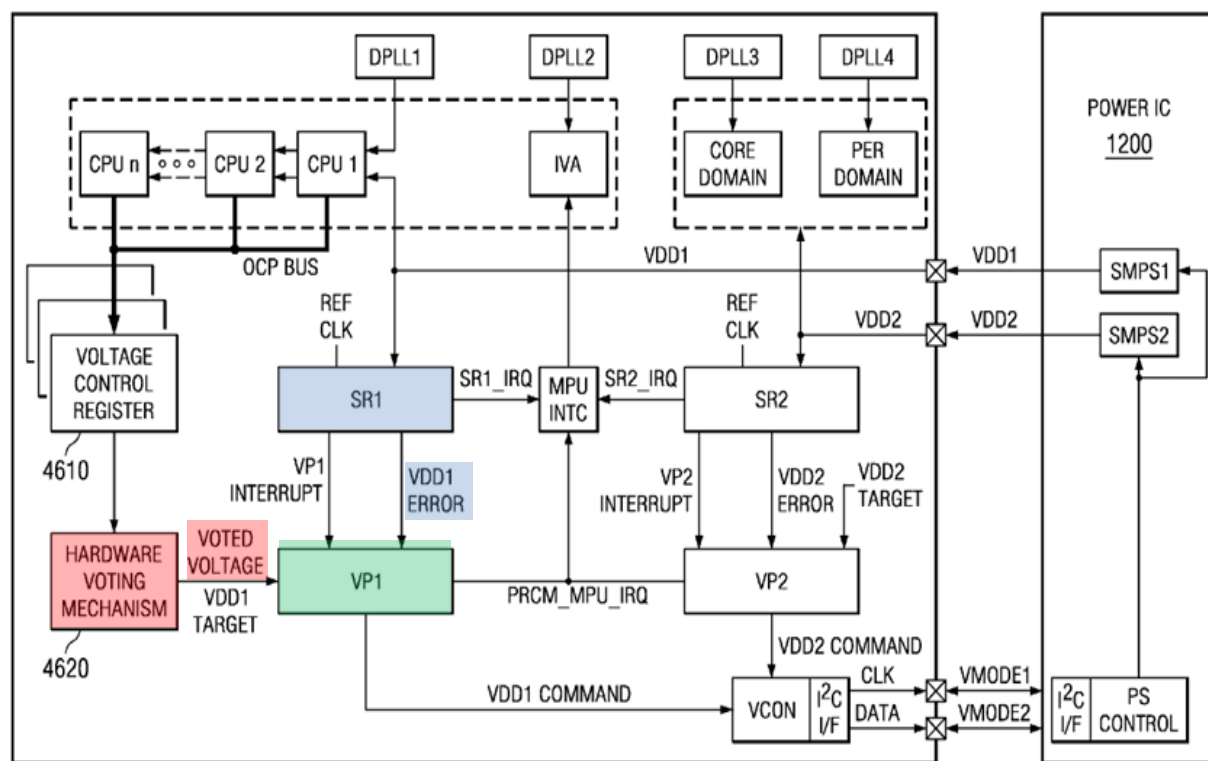


FIG. 10

165. As explained above for Element 1[e3], Voltage Processor VP1 generates VDD1 Command based on VDD1 Error (workload configuration input) and the voted voltage (global EPB value). *See supra* Section VII.B.1.f.iii. It would have been obvious to a POSITA that Voltage Processor 1 indexes the lookup table in Titiano by those two variables to retrieve the VDD1 Command. A two-dimensional table is a well-known, fast, and efficient way to retrieve a result that is a function of two variables. *See supra* Section VI.C. Thus, a POSITA would have been motivated to implement the lookup table as such a two-dimensional table. In such an implementation, the power controller (which includes Voltage Processors

VP1) would access the lookup table based on workload configuration input (VDD1 Error) and the global EPB (voted voltage) and use the information from the table to update at least one power management setting (VDD1 Command).

166. Accordingly, Titiano's implementation of AVS and DVFS teaches claim 2.

- b. Claim 3: The processor of claim 2, wherein the tuning circuit is to update the at least one setting of the at least one power management feature responsive to the global EPB value.**

167. Titiano teaches dependent claim 3 because claim 3 is encompassed by the limitations of claims 1 and 2. *See supra* Section VII.B.1–2.

- c. Claim 4: The processor of claim 3, wherein the tuning circuit includes a sampler to receive the plurality of EPB values from at least some of the plurality of cores and a combiner to generate the global EPB value from the EPB value from the plurality of cores.**

168. Titiano teaches dependent claim 4. As explained above for Element 1[e2], the tuning circuit in Titiano includes registers which receive the plurality of EPB values from the plurality of cores. *See supra* Section VII.B.1.f.ii. These registers correspond to the claimed “sampler.”

169. Also, as explained above for Element 1[e2], the tuning circuit in Titiano includes an arbitrating circuit that generates the global EPB (maximum voted voltage) from the EPB values from the cores. This arbitrating circuit is the

claimed “combiner.” *See supra* Section VII.B.1.f.ii.

170. Thus, the tuning circuit in Titiano includes a sampler (registers) to receive the plurality of EPB values (voltage requests) from at least some of the plurality of cores and a combiner (arbitrating circuit) to generate the global EPB value (maximum voted voltage) from the EPB value from the plurality of cores.

171. Accordingly, Titiano teaches dependent claim 4.

d. Claim 5: The processor of claim 4, wherein the sampler is to receive the EPB value from a plurality of threads to execute on the at least some of the plurality of cores.

172. Titiano teaches dependent claim 5. As explained above for claim 4, Titiano teaches a sampler (registers) that receives EPB values from the plurality of cores. *See supra* Section VII.B.2.c. Claim 5 adds only that the EPB values are “from a plurality of threads” on the cores. A POSITA would have understood that the term “thread” refers to the software thread running on the core. This understanding is consistent with the specification of the ’197 Patent, which describes “logical threads.” *See* Ex-1001, 4:18–25. Titiano describes “symmetric multiprocessing (SMP) core(s) for use in the systems of FIGS. 1-19.” Ex-1007 ¶221. The plurality of SMP cores in Titiano would include multiple threads. Thus, the sampler in Titiano receives the EPB values “from a plurality of threads to execute on the at least some of the plurality of cores.”

173. Alternatively, it would have been obvious to a POSITA that the SMP cores in Titiano would include a plurality of threads and thus that the sampler in Titiano receives the EPB values “from a plurality of threads to execute on the at least some of the plurality of cores.”

174. Accordingly, Titiano teaches dependent claim 5.

- e. **Claim 6: The processor of claim 4, wherein the power controller is to generate a bin value from the global EPB value and to access the power-performance tuning table using the bin value.**

175. Titiano teaches dependent claim 6. According to Titiano, Operating Performance Points (OPPs) (Opp1, Opp2, etc.) are associated with different voltage and frequency pairs. Ex-1007 ¶109 (“DVFS pairs [v]oltage and frequency to form OPPs (operating performance points) of different predetermined voltage and frequency where $OPP_n = [V_n, F_n]$.”). Thus, Titiano teaches generating a bin value (OPP number) from the global EPB value (max voted voltage). It would have been obvious to a POSITA that the power-performance tuning table could be accessed by the global EPB value (max voted voltage) or the bin value (OPP number).

176. Accordingly, Titiano teaches dependent claim 6.

- f. Claim 19: The processor of claim 1, wherein the power controller is to enable a first power management feature based on the updated at least one setting.**

177. Titiano discloses dependent claim 19. As described above with respect to Element 1[e3], the tuning circuit in Titiano “updat[es] at least one setting of at least one of the plurality of power management features.” *See supra* Section VII.B.1.f.iii. The updated setting relates to VDD1 supply voltage. *Id.* As explained above, by updating VDD1 supply voltage, the power controller in Titiano enables a power management feature called Dynamic Voltage Frequency Scaling (DVFS), which scales the supply voltage to the lowest voltage to deliver various operation modes and frequencies according to predicted needs. Ex-1007 ¶107; *see also id.* ¶¶109, 111, 112, 122, 126.

178. Accordingly, Titiano teaches dependent claim 19.

- g. Claim 20: The processor of claim 1, wherein the power controller is to send an update message to a destination agent of the processor to cause the destination agent to control a first power management feature based on the updated at least one setting.**

179. Titiano teaches dependent claim 20. As explained above for claim 19, the power controller in Titiano enables a first power management feature (DVFS) based on the updated at least one setting (VDD supply voltage). *See supra* Section VII.B.2.f. Claim 20 is similar to claim 19 but further adds the use of a “destination

agent.”

180. As explained above, Titiano teaches the limitations of claim 1. *See supra* Section VII.B.1. With respect to claim 20, the “voted resource level” corresponds to the claimed “update message,” and the “scalable resource controller” corresponds to the claimed “destination agent of the processor.”

181. Titiano discloses the use of a destination agent, such as scalable resource controller 4540 in Figure 8, scalable voltage regulator 4640 in Figure 9, and Power IC 1200 in Figures 10, 12, 13, and 19.

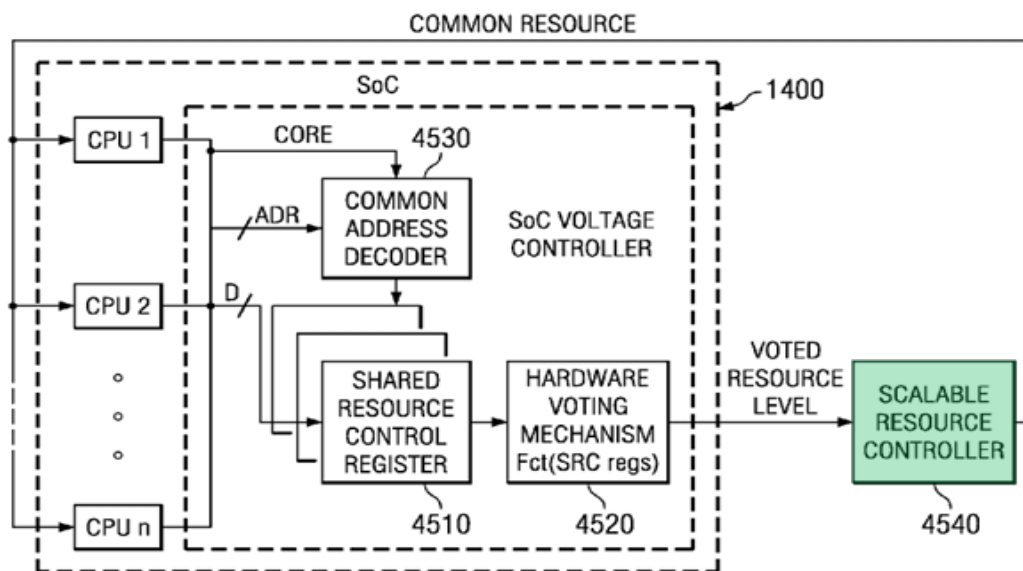


FIG. 8

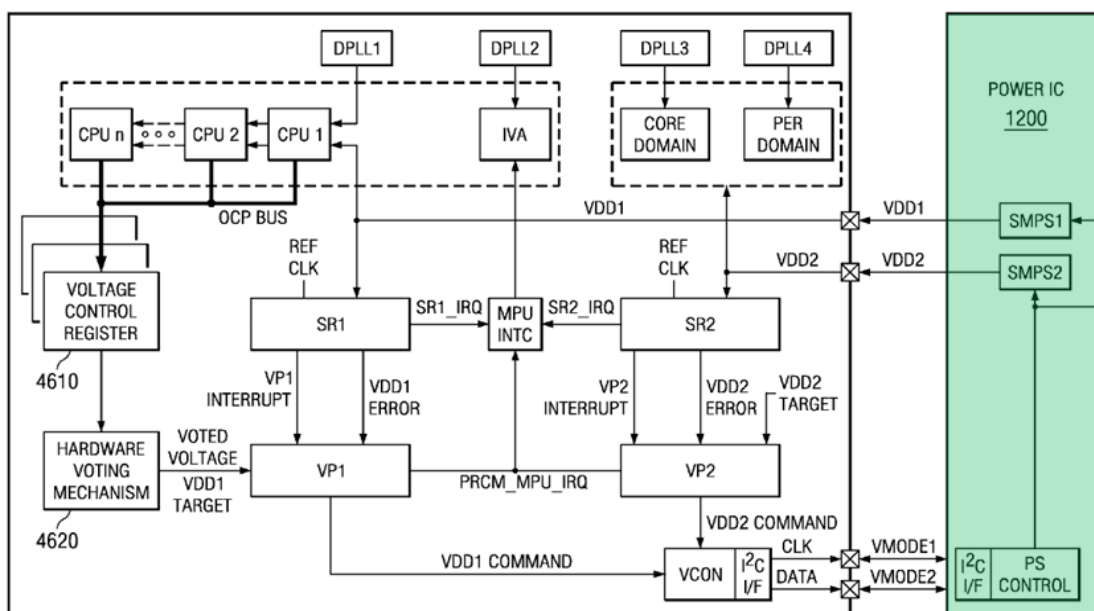


FIG. 10

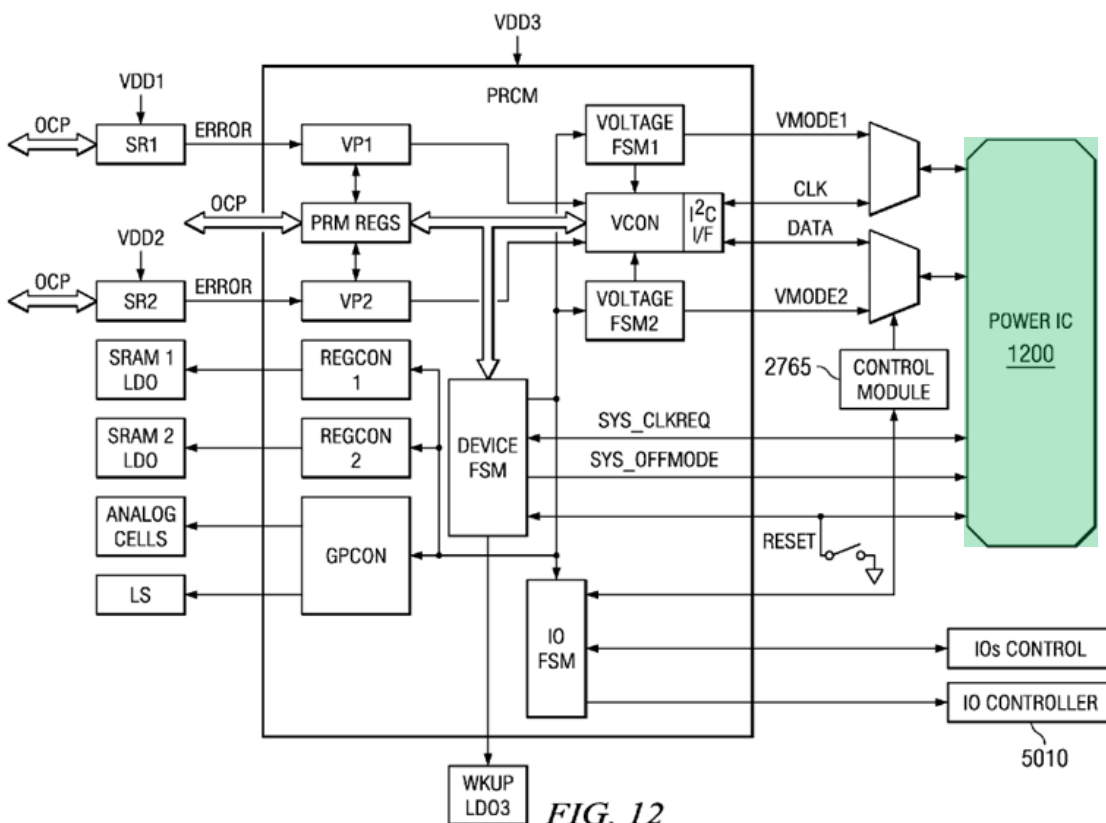


FIG. 12

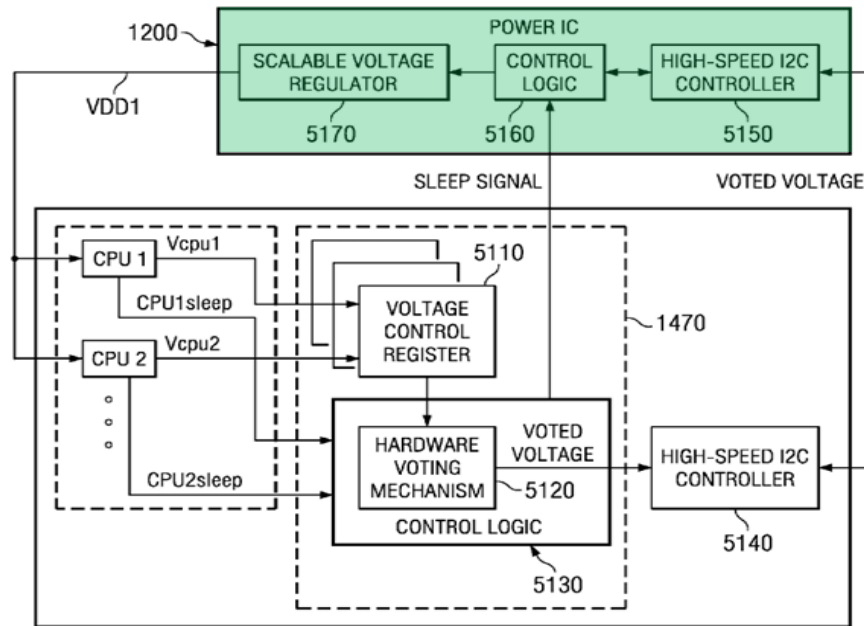


FIG. 13

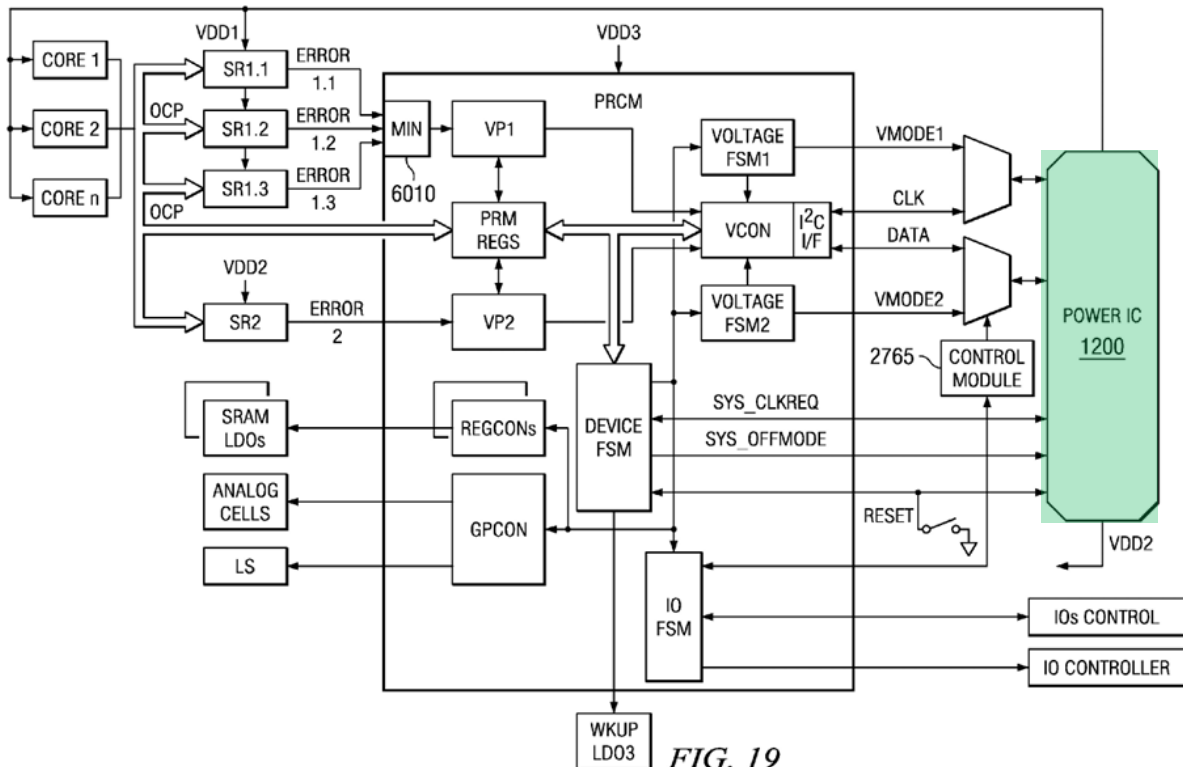


FIG. 19

182. The power controller in Titiano sends an update message to these

destination agents to control a power management feature (DVFS) based on the updated VDD voltage setting. Ex-1007 ¶¶136, 138 (“Hardware voting mechanism 4520 then automatically delivers a command or Voted Resource Level to scale the voltage as specified by that function ... or includes the circuitry 4540 itself to scale the voltage to the highest request voltage.”); *id.* ¶155 (In Figs. 10 and 12, “The SMPS register supplies an SMPS voltage command to the Voltage Controller VCON which in turn communicates via I2C to Power IC 1200 to adjust the voltage to the commanded voltage.”). As explained above for Element 1[e3], the power controller in Figures 10 and 12 of Titiano communicates the voltage setting to Power IC 1200 as Data over the I2C interface. *See supra* Section VII.B.1.f.iii.

183. Accordingly, Titiano teaches dependent claim 20.

3. Independent Claim 11

- a. **Element 11[pre]: A non-transitory machine-readable medium having stored thereon instructions, which if performed by a machine cause the machine to perform a method comprising:**

184. To the extent that the preamble of claim 11 is limiting, Titiano discloses the preamble. Titiano states that its power management process may be implemented by hardware, software, or firmware. Ex-1007 ¶258; *see also id.* ¶115 (“DVFS hardware and software”), *id.* ¶116 (“DVFS software”), *id.* ¶119 (“power management software running on one or more of the processors,”); *id.* (“DVFS

voltage scaling software or hardware”), *id.* ¶211 (“The deployment of DPS is supported by suitable performance prediction software and/or hardware.”).

- b. Element 11[a]: receiving an energy performance bias (EPB) value in a tuning circuit of a power control logic of a processor from a plurality of threads executing on the processor and determining a global EPB value based at least in part on the EPB value received from the plurality of threads, the EPB value to indicate a preference of a user for a tradeoff between power optimization and performance optimization;**

185. Titiano discloses Element 11[a]. As described above with reference to Element 1[e2], Titiano teaches a power controller (which is the same as “power control logic”) that receives EPB values and determines a global EPB based on the EPB values, and the EPB values indicate a tradeoff between power optimization and performance optimization. *See supra* Section VII.B.1.f.ii.

186. Element 11[a] adds only that the EPB value “indicate[s] a preference of a user” for the power-performance tradeoff. Titiano also teaches this further limitation. The voted voltages in Titiano are programmed into the voltage control registers via software operating on the CPU cores. Ex 1007 ¶119. Titiano suggests that a user may also control the software to effectuate the programming of the voted voltages in the voltage control registers. *See id.* ¶135 (“When performing DVFS, the user can program a new operating point in the sensor module by programming the new frequency target and caus[ing] the sensor module to re-

calculate the error.”). Regardless of whether the software is acting on behalf of a user, the OS, or the CPU cores, the software performs the same function (i.e., determining a required voltage for the CPU cores and programming it in the voltage control registers). Thus, it would have been obvious to enable a user to control the software in Titiano to determine an EPB value indicating the user’s preference for a tradeoff between power and performance optimization.

- c. Elements 11[b]: accessing a table based on the global EPB value; and
Element 11[c]: updating at least one setting of a power management feature controlled by the power control logic based on information obtained from the table.**

187. Elements 11[b] and 11[c] are the same subject matter as recited in claim 2. Thus, Titiano discloses Elements 11[b] and 11[c] for the same reasons described above for claim 2. *See supra* Section VII.B.2.a.

- 4. Dependent Claim 13: The non-transitory machine-readable medium of claim 11, wherein the method further comprises providing an interface to enable a user, an operating system or a baseboard management controller to set the EPB value, and controlling a plurality of power management features via input of the EPB value.**

188. Titiano teaches claim 13. Titiano describes that “[w]hen performing DVFS, the user can program a new operating point in the sensor module by programming the new frequency target and caus[ing] the sensor module to recalculate the error.” Ex-1007 ¶135. Titiano also refers to the sensor module as an

“interface.” *Id.* ¶153 (“The error is output on ***SR interface*** and is related to the actual voltage step that needs to be applied.”) (emphasis added). Thus, Titiano teaches “controlling a plurality of power management features via input of the EPB value.”

189. As explained above for claim 1, Titiano teaches “a power controller to control a plurality of power management features of the processor” using EPB values. *Supra* Sections VII.B.1.e and VII.B.1.f.ii. Thus, Titiano teaches “providing an interface to enable a user, an operating system or a baseboard management controller to set the EPB value, and controlling a plurality of power management features via input of the EPB value.”

C. Ground 4: Claims 2 and 15 Are Obvious Over Titiano in View of Hansquine

1. A POSITA would have been motivated to combine Titiano and Hansquine

190. In addition to the reasons set out below, a POSITA would have been motivated to combine Titiano with Hansquine’s teachings, and would have had a reasonable expectation of success in doing so, because they are from the same field of endeavor and relate to the same well-known technologies. Titiano and Hansquine provide complementary techniques for power management of multi-core processors using a power management feature known as Adaptive Voltage Scaling (AVS). Ex-1007 ¶¶125, 134–35, 160; Ex-1008 ¶¶ 3, 11, 37. A POSITA

seeking to improve power management and efficiency in a multi-core processor using AVS would have known of and consulted each of these proposed techniques.

191. Titiano teaches that AVS may be used to generate “an error output that indicates whether to raise or lower the voltage.” Ex-1007 ¶153. This error output indicates “the temperature induced current performance of the device.” *Id.* ¶134. Titiano further describes that “[e]ach of the Voltage Processors contain Look-Up Tables and [are] correctly initialized and configured to optimize the voltage control.” *Id.* ¶159. Hansquine provides a two-dimensional “AVS frequency/voltage level table providing voltage level values as a function of operating frequencies and temperature level bands of a functional circuit(s).” Ex-1008 ¶29. Thus, Hansquine provides additional implementation details that could be used to implement AVS in Titiano.

192. A POSITA would have had a reasonable expectation of success in combining the teachings of Titiano and Hansquine because each purports to implement AVS in similar and predictable ways (using a lookup table)—for example, by using well-known implementations, such as using AVS to manage power, and techniques such as providing a two-dimensional look-up table to optimize voltage control.

2. Dependent Claim 2

193. Titiano in view of Hansquine teaches dependent claim 2. As described

above for Ground 3, it would have been obvious to implement the AVS lookup table in Titiano as a two-dimensional table. *See supra* Section VII.B.2.a.

Hansquine describes implementing an AVS lookup table as two-dimensional table, as shown in Figure 13.

AVS Frequency/Voltage Level Table (140)

142

	T (°C)		f ₁	f ₂	...	f _N
B ₁ →	80	60	1.2V	1.18V	...	
B ₂ →	60	40	1.1V	1.08V	...	
B ₃ →	40	20	1.0V	.99V	...	
B ₄ →	20	0	⋮	⋮	...	⋮
B _N →	⋮	⋮	.95V	.94V	...	

144

$f_1 > f_2 > f_N$

FIG. 13

Thus, the Titiano-Hansquine combination further supports that it would have been obvious to implement the AVS lookup table in Titiano as a two-dimensional table.

194. A POSITA would have been motivated to combine the teachings of Titiano and Hansquine because Hansquine provides additional implementation details consistent with the teachings of Titiano. Both references use Adaptive Voltage Scaling techniques to optimize power consumption in a multi-core

processor. Ex-1007 ¶111; Ex-1008 ¶3. Both references also describe the use of a lookup table to implement AVS. Ex-1007 ¶159; Ex-1008 ¶74. Thus, a POSITA would have also been motivated to combine these teachings of Titiano and Hansquine because the table in Hansquine provides a responsive and efficient way to determine the VDD1 voltage command in Titiano.

195. In view of Hansquine's two-dimensional AVS lookup table, it would have been obvious to a POSITA to implement the AVS lookup table in Titiano as a two-dimensional table. *See supra* Section VI.C. In such an implementation, the power controller (which includes Voltage Processors VP1) would access the lookup table based on workload configuration input (VDD1 Error) and the global EPB value (voted voltage) and use the information from the table to update at least one power management setting (VDD1 Command).

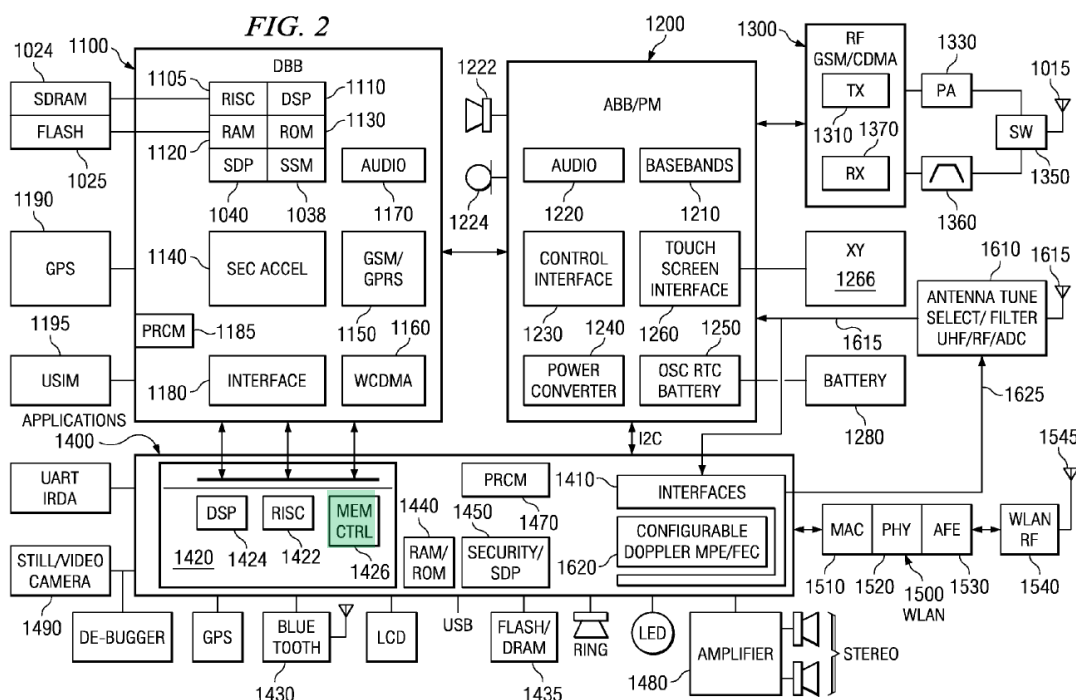
196. Accordingly, the Titiano-Hansquine combination teaches claim 2.

3. Independent Claim 15

a. Element 15[a]: a multicore processor including a plurality of cores, at least one cache memory and a memory controller;

197. Titiano discloses Element 15[a]. As explained above for Elements 1[pre], 1[a], and 1[b], Titiano discloses “a multicore processor including a plurality of cores” and “at least one cache memory.” *See supra* Section VII.B.1.a–c. Claim 15 adds only “a memory controller.” As shown below in Figure 2, the processor in

Titiano includes a memory controller 1426. Ex-1007, ¶¶52, 64.



Titiano describes that “[o]n chip 1400, the *shared memory controller 1426* in circuitry 1420 interfaces the RISC processor 1420 and the DSP 1424 via an on-chip bus to on-chip memory 1440 with RAM and ROM.)” *Id.* ¶52.

198. Accordingly, Titiano discloses Element 15[a].

- b. Element 15[b]: wherein the tuning circuit is to access an entry of a tuning table based at least in part on the EPB value and a workload configuration value and update a setting for a power management feature responsive to a value stored in a field of the entry;**

199. Element 15[b] is the same as claim 2, except Element 15[b] refers to “a value stored in a field of the entry” in contrast to Claim 2’s reference to “information” from the tuning table. Although Claim 15 refers to workload

configuration “value” and claim 2 refers to workload configuration “input,” these terms do not appear to have different meanings. In any case, the workload configuration input in Titiano is a value. *See supra* Section VII.B.1.f.i. Because Titiano and Hansquine disclose a tuning table with values (i.e., voltage-frequency pairs,) the Titiano-Hansquine combination teaches Element 15[b] for the same reasons described above for claim 2. *See supra* Section VII.C.2.

- c. Element 15[c]: wherein the workload configuration value is to indicate a predominant workload type to be executed on the system, the tuning table including a first set of entries associated with a first workload type and a second set of entries associated with a second workload type;**

200. The Titiano-Hansquine combination teaches Element 15[c]. As explained above with reference to Element 1[e1], the VDD1 Error signal in Titiano indicates “the temperature induced current performance of the device” (Ex-1007 ¶134) and thus is a workload configuration input (or value). *See supra* Section VII.B.1.f.i. The workload configuration value indicates a predomination workload type. If the workload type is processor-intensive, a higher “temperature induced current performance of the device” results in a corresponding value for the VDD1 Error signal. If the workload is less processor-intensive, a lower “temperature induced current performance of the device” results in a corresponding value for the VDD1 Error signal. Thus, the workload configuration value indicates “a

predominant workload type” (processor-intensive vs. less processor-intensive).

201. As explained above for claim 2, the Titiano-Hansquine combination teaches that the AVS table in Titiano could be implemented as a two-dimensional table, which is indexed by a workload configuration value (VDD1 Error signal). *See supra* Section VII.C.2. In such an embodiment, the table would include entries indexed by a VDD1 Error signal, such that a different entries would be utilized depending on the value of the VDD1 Error signal. Thus, for at least two values of the VDD1 Error signal, a first set of entries would be associated with a first workload type (processor-intensive workloads), and a second set of entries would be associated with a second workload type (less processor-intensive workloads).

d. Element 15[d]: a system memory coupled to the multicore processor via a memory interconnect.

202. Titiano discloses Element 15[d]. Titiano states, “[o]n chip 1400, the shared memory controller 1426 in circuitry 1420 interfaces the RISC processor 1420 and the DSP 1424 via an on-chip bus to on-chip memory 1440 with RAM and ROM.” Ex-1007 ¶64. Thus, Titiano describes a system memory (chip memory 1440) coupled to the multicore processor (RISC processor 1420 and the DSP 1424) via a memory interconnect (the on-chip bus).

- 4. Dependent Claim 16: The system of claim 15, wherein the tuning circuit is to update at least one setting of a first power management feature controlled by a power control logic of the multicore processor, the at least one setting obtained from a field of an entry of the tuning table associated with a bin in which the EPB value is included.**

203. Titiano teaches dependent claim 16. As explained above with respect to Claim 1, Titiano discloses a “power controller includ[ing] a tuning circuit ... to update at least one setting of at least one of the plurality of power management features.” *See supra* Section VII.B.1.f. Titiano further discloses that the power controller (e.g., Power Management Control Module (PCRM)) includes Control Logic. *See, e.g.*, Ex-1007 ¶¶185–188, Fig. 13. Thus, a POSITA would have understood that the “at least one setting of a first power management feature” is “controlled by a power control logic of the multicore processor,” as recited in claim 16.

204. In addition, as explained with respect to claim 6, Titiano discloses that the “power controller is to generate a bin value from the global EPB value and to access the power-performance tuning table using the bin value.” *See supra* Section VII.B.2.e. Thus, Titiano also discloses that “the at least one setting [is] obtained from a field of an entry of the tuning table associated with a bin in which the EPB value is included,” as recited in claim 16.

5. Dependent Claim 17: The system of claim 16, wherein multicore processor comprises the tuning circuit.

205. Titiano teaches dependent claim 17. Titiano teaches that the CPU cores and the tuning circuit (e.g., the shared resource control register and hardware voting mechanism) are located on the same processor. *See, e.g.*, Ex-1007 ¶137; Figs. 8–9; *supra* Section VII.B.1. Thus, the multicore processor of Titiano comprises the tuning circuit, as recited in claim 17.

6. Dependent Claim 18: The system of claim 15, wherein the multicore processor comprises a power control logic, the power control logic comprising the tuning circuit, wherein the tuning circuit is to generate a bin value from the EPB value and access the tuning table using the bin value.

206. Titiano teaches dependent claim 18. The claimed “power control logic” corresponds to the tuning circuit, which is the portion of PRCM 1470 that performs the functions recited in Elements 1[e1], 1[e2] and 1[e3] (*see supra* Section VII.B.1.e). Titiano further teaches that the tuning circuit (e.g., portion of Power Management Control Module (PCRM) implementing the voltage voting process) includes Control Logic. *See, e.g.*, Ex-1007 ¶¶185–188, Fig. 13; *supra* Section VII.C.4. Thus, a POSITA would have understood that “the power control logic compris[es] the tuning circuit,” as recited in claim 18.

207. As explained above for claim 6, the tuning circuit generates a bin value from the EPB value and accesses the tuning table using the bin value. *See*

supra Section VII.B.2.e. Thus, Titiano teaches that “the tuning circuit ... generate[s] a bin value from the EPB value and access[es] the tuning table using the bin value.”

VIII. CONCLUSION

208. For at least these reasons, Claims 1–6, 11, 13, and 15–20 of the ’197 Patent are invalid over the prior art.

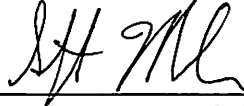
209. In signing this declaration, I understand that the declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I acknowledge that I may be subject to cross-examination in this case. If cross-examination is required of me, I will appear for cross-examination during the time allotted for cross-examination.

210. I declare that all statements made herein of my knowledge are true, that all statements made on information and belief are believed to be true, and that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

211. I declare under penalty of perjury that the foregoing is true and correct.

Respectfully Submitted,

March 24, 2023

A handwritten signature in black ink, appearing to read 'S. W. Melvin', written over a horizontal line.

Dr. Stephen W. Melvin