UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

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

v.

DAEDALUS PRIME LLC,
Patent Owner.

**DECLARATION OF TREVOR MUDGE
IN SUPPORT OF PETITION FOR *INTER PARTES* REVIEW OF
U.S. PATENT NO. 8,775,833**

I. INTRODUCTION 1
A. Qualifications 1
B. Materials Considered 3
II. LEGAL STANDARDS 4
A. Anticipation 5
B. Obviousness 6
C. Level of Ordinary Skill in the Art 9
D. Claim Construction 10
III. SUMMARY OF GROUNDS 11
IV. THE ’833 PATENT 12
A. Overview of the ’833 Patent 12
B. Prosecution History of the ’833 Patent 15
V. TECHNOLOGY BACKGROUND 15
A. Power, voltage, and frequency control approaches 17
B. Multi-core processing approaches 17
C. NorthBridge – SouthBridge Architecture 18
VI. THE PRIOR ART IN THE APPLIED INVALIDITY GROUNDS 19
A. Finkelstein 19
B. Conroy 21
C. Nussbaum 23
D. Bose 25
VII. CLAIM CONSTRUCTION 26
VIII. DETAILED EXPLANATION OF THE UNPATENTABILITY GROUNDS 27
A. Ground 1: Claims 1-5 and 7 are rendered obvious by Finkelstein. 27
1. Independent claim 1 27
a. Element 1[pre]: A processor comprising: 27

b. Element 1[A]: a first domain and a second domain, each of the first and second domains to operate at an independent voltage and frequency; 28

c. Element 1[B]: a memory controller coupled to the first and second domains; 31

d. Element 1[C]: at least one interface; and 33

e. Element 1[D.i]: first logic to dynamically allocate a power budget for the processor between the first and second domains at run time 35

f. Element 1[D.ii]: according to a first sharing policy value for the first domain and a second sharing policy value for the second domain, the first and second sharing policy values controllable by user-level software. 37

2. Dependent claims 2-5 and 7 40

a. Claim 2: The processor of claim 1, wherein the first logic is to determine a portion of the power budget to allocate to the first domain based on the first sharing policy value. 40

b. Claim 3: The processor of claim 1, wherein the first logic is to dynamically allocate the power budget further according to a first minimum reservation value for the first domain and a second minimum reservation value for the second domain, the first and second minimum reservation values controllable by the user-level software.....41

c. Claim 4: The processor of claim 3, wherein the first logic is to provide at least a first portion of the power budget to the first domain, the first portion corresponding to the first minimum reservation value. 43

Petitioner Mercedes Ex-1002, 0003

d. Claim 5: The processor of claim 1, wherein the first logic is to determine the power budget for a current time interval based at least in part on a power budget carried forward from a previous time interval, a power consumed in the previous time interval, and a power budget decay value. 44

e. Claim 7: The processor of claim 1, wherein the first logic is to dynamically allocate substantially all of the power budget to the first domain for a first workload, and to dynamically allocate substantially all of the power budget to the second domain for a second workload executed after the first workload. 46

B. Ground 2: Claim 7 is rendered obvious by Finkelstein in further view of Conroy. 49

1. Dependent claim 7 49

a. Claim 7: The processor of claim 1, wherein the first logic is to dynamically allocate substantially all of the power budget to the first domain for a first workload, and to dynamically allocate substantially all of the power budget to the second domain for a second workload executed after the first workload. 49

C. Ground 3: Claims 13-15 and 17-18 are rendered obvious by Nussbaum. 51

1. Independent claim 13 51

a. Element 13[Pre]: A system comprising:.....51

b. Element 13[A.i]: a multicore processor having a first domain including a plurality of cores, a second domain including a graphics engine, and 52

c. Element 13[A.ii(1)]: a third domain including system agent circuitry, ...55

d. Element 13[A.ii(2)]: the third domain to operate at a fixed power budget and.....58

e. Element 13[A.ii(3)]: including a power sharing logic to dynamically allocate a variable power budget between the first and second domains 59

f. Element 13[A.ii(4)]: based at least in part on a first power sharing value for the first domain stored in a first storage and a second power sharing value for the second domain stored in a second storage; and.....61

g. Element 13[B]: a dynamic random access memory (DRAM) coupled to the multicore processor. 63

2. Dependent claims 14-15 and 17-18 64

a. Claim 14: The system of claim 13, wherein the power sharing logic to determine the variable power budget for a current time interval and to allocate a first portion of the variable power budget to the first domain according to the first power sharing value, and to allocate a second portion of the variable power budget to the second domain according to the second power sharing value.....64

b. Claim 15: The system of claim 14, wherein the power sharing logic is to dynamically allocate substantially all of the variable power budget to the first domain for a first workload, and to dynamically allocate substantially all of the variable power budget to the second domain for a second workload executed after the first workload.....65

c. Claim 17: The system of claim 14, wherein the power sharing logic is to further allocate the first portion of the variable power budget according to a first minimum reservation value for the first domain and allocate the second portion of the variable power budget according to a second minimum reservation value for the second domain. 68

d. Claim 18: The system of claim 14, wherein the power sharing logic is to further allocate the variable power budget according to a preference value, the preference value to favor the second domain over the first domain.....70

D. Ground 4: Claim 16 is rendered obvious by Nussbaum in view of Bose. 72

1. Dependent claim 16 72

a. Claim 16: The system of claim 14, wherein the power sharing logic is to increment the first power sharing value when a request for a higher frequency for the first domain is not granted, and to increment the second power sharing value when a request for a higher frequency for the second domain is not granted. 72

E. Ground 5: Claims 13-15 and 17-18 are rendered obvious by Conroy. 76

1. Independent claim 13 76

a. Element 13[Pre]: A system comprising: 76

b. Element 13[A.i]: a multicore processor having a first domain including a plurality of cores, a second domain including a graphics engine, and 76

c. Element 13[A.ii(1)]: a third domain including system agent circuitry, ... 80

d. Element 13[A.ii(2)]: the third domain to operate at a fixed power budget and 82

e. Element 13[A.ii(3)]: including a power sharing logic to dynamically allocate a variable power budget between the first and second domains 82

f. Element 13[A.ii(4)]: based at least in part on a first power sharing value for the first domain stored in a first storage and a second power sharing value for the second domain stored in a second storage; and 83

g. Element 13[B]: a dynamic random access memory (DRAM) coupled to the multicore processor. 87

2. Dependent claims 14-15 and 17-18 88

a. Claim 14: The system of claim 13, wherein the power sharing logic to determine the variable power budget for a current time interval and to allocate a first portion of the variable power budget to the first domain according to the first power sharing value, and to allocate a second portion of the variable power budget to the second domain according to the second power sharing value. 88

b. Claim 15: The system of claim 14, wherein the power sharing logic is to dynamically allocate substantially all of the variable power budget to the first domain for a first workload, and to dynamically allocate substantially all of the variable power budget to the second domain for a second workload executed after the first workload. 89

c. Claim 17: The system of claim 14, wherein the power sharing logic is to further allocate the first portion of the variable power budget according to a first minimum reservation value for the first domain and allocate the second portion of the variable power budget according to a second minimum reservation value for the second domain. 90

d. Claim 18: The system of claim 14, wherein the power sharing logic is to further allocate the variable power budget according to a preference value, the preference value to favor the second domain over the first domain. 91

F. Ground 6: Claim 16 is rendered obvious by Conroy in view of Bose. 92

1. Dependent claim 16 92

a. Claim 16: The system of claim 14, wherein the power sharing logic is to increment the first power sharing value when a request for a higher frequency for the first domain is not granted, and to increment the second power sharing value when a request for a higher frequency for the second domain is not granted. 92

Petitioner Mercedes Ex-1002, 0007

G. Ground 7: Claims 1-5, 7, 13-15, and 17-18 are rendered obvious by Conroy in view of Finkelstein. 94

1. A POSITA would be motivated to combine the teachings of Conroy and Finkelstein, and would have a reasonable expectation of success in doing so. 94

2. Independent claim 1 96

a. Element 1[Pre]: A processor comprising: 96

b. Element 1[A]: a first domain and a second domain, each of the first and second domains to operate at an independent voltage and frequency; 98

c. Element 1[B]: a memory controller coupled to the first and second domains; 102

d. Element 1[C]: at least one interface; and 105

e. Element 1[D.i]: first logic to dynamically allocate a power budget for the processor between the first and second domains at run time 106

f. Element 1[D.ii]: according to a first sharing policy value for the first domain and a second sharing policy value for the second domain, the first and second sharing policy values controllable by user-level software. 107

3. Dependent claims 2-5 and 7 109

a. Claim 2: The processor of claim 1, wherein the first logic is to determine a portion of the power budget to allocate to the first domain based on the first sharing policy value. 109

b. Claim 3: The processor of claim 1, wherein the first logic is to dynamically allocate the power budget further according to a first minimum reservation value for the first domain and a second minimum reservation value for the second domain, the first and second minimum reservation values controllable by the user-level software 110

c. Claim 4: The processor of claim 3, wherein the first logic is to provide at least a first portion of the power budget to the first domain, the first portion corresponding to the first minimum reservation value. 111

d. Claim 5: The processor of claim 1, wherein the first logic is to determine the power budget for a current time interval based at least in part on a power budget carried forward from a previous time interval, a power consumed in the previous time interval, and a power budget decay value. 112

e. Claim 7: The processor of claim 1, wherein the first logic is to dynamically allocate substantially all of the power budget to the first domain for a first workload, and to dynamically allocate substantially all of the power budget to the second domain for a second workload executed after the first workload. 114

4. Independent claim 13 114

a. Element 13[Pre]: A system comprising: 114

b. Element 13[A.i]: a multicore processor having a first domain including a plurality of cores, a second domain including a graphics engine, and 114

c. Element 13[A.ii(1)]: a third domain including system agent circuitry, ... 114

d. Element 13[A.ii(2)]: the third domain to operate at a fixed power budget and 114

e. Element 13[A.ii(3)]: including a power sharing logic to dynamically allocate a variable power budget between the first and second domains 115

f. Element 13[A.ii(4)]: based at least in part on a first power sharing value for the first domain stored in a first storage and a second power sharing value for the second domain stored in a second storage; and 115

g. Element 13[B]: a dynamic random access memory (DRAM) coupled to the multicore processor. 115

Petitioner Mercedes Ex-1002, 0009

5. Dependent claims 14-15 and 17-18 115

a. Claim 14: The system of claim 13, wherein the power sharing logic to determine the variable power budget for a current time interval and to allocate a first portion of the variable power budget to the first domain according to the first power sharing value, and to allocate a second portion of the variable power budget to the second domain according to the second power sharing value.....115

b. Claim 15: The system of claim 14, wherein the power sharing logic is to dynamically allocate substantially all of the variable power budget to the first domain for a first workload, and to dynamically allocate substantially all of the variable power budget to the second domain for a second workload executed after the first workload.....116

c. Claim 17: The system of claim 14, wherein the power sharing logic is to further allocate the first portion of the variable power budget according to a first minimum reservation value for the first domain and allocate the second portion of the variable power budget according to a second minimum reservation value for the second domain. ...116

d. Claim 18: The system of claim 14, wherein the power sharing logic is to further allocate the variable power budget according to a preference value, the preference value to favor the second domain over the first domain.....116

IX. CONCLUSION.....116

I. INTRODUCTION

1. I have been retained by Samsung Electronics Co., Ltd. and Samsung Electronics America, Inc. (collectively, "Samsung" or "Petitioner"), as an independent expert in this proceeding before the Patent Trial and Appeal Board ("PTAB" or "Board"). I understand that Samsung is requesting that the Board institute an *inter partes* review ("IPR") proceeding of U.S. Patent No. 8,775,833 ("the '833 Patent") (Ex-1001), currently assigned to Daedalus Prime LLC ("Patent Owner" or "PO").

2. I am not and have never been an employee of Samsung. I am being compensated at my usual and customary rate of $600 per hour. No part of my compensation depends on the outcome of this proceeding, and I have no other interest in this proceeding.

3. I have been asked to provide my independent analysis of the '833 Patent in light of the prior art publications cited below. I have also been asked to consider the state of the art and prior art available as of September 6, 2011. Based on the prior art discussed in this declaration, it is my opinion that Claims 1-5, 7, and 13-18 of the '833 Patent are unpatentable for the reasons provided below.

A. Qualifications

4. I am currently a the Bredt Family Professor of Computer Science and Engineering at the University of Michigan, Ann Arbor. All of my opinions stated in

this declaration are based on my own personal knowledge and professional judgment. In forming my opinions, I have relied on my over 45 years of research, academic, industry, and consulting engineering experience in IC (integrated circuit) processing, semiconductor devices, and computer architecture with an emphasis on power and energy control.

5. I am over 18 years of age and, if I am called upon to do so, I would be competent to testify as to the matters set forth herein. I understand that a copy of my current curriculum vitae, which details my education and professional and academic experience, is being submitted by Petitioner as Exhibit 1003. The following provides an overview of some of my experience that is relevant to the matters set forth in this declaration.

6. I received the Ph.D. in Computer Science from the University of Illinois, Urbana. I am the Bredt Family Professor of Computer Science and Engineering at the University. I am the author of numerous papers on computer architecture, programming languages, VLSI design, and computer vision. I have chaired 57 Ph.D. theses in these areas. In 2014 I received the ACM/IEEE CS Eckert-Mauchly Award for "pioneering contributions to low-power computer architecture and its interaction with technology." This is known as the computer architecture community's most prestigious award. I also received the University of Illinois

Distinguished Alumni Award. I am a Life Fellow of the IEEE, a Fellow of the ACM, and a member of the IET and the British Computer Society.

7. Based on my experience and education, I believe that I am qualified to opine as to the knowledge and level of skill of one of ordinary skill in the art at the time of the alleged invention of the '833 Patent, as well as the state of the art at that time.

B. Materials Considered

8. In forming my opinions, I have reviewed the following documents:[1]

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

[1] Four-digit pin citations that begin with 0 are to the branded numbers 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-1013	INTENTIONALLY LEFT BLANK
Ex-1014	INTENTIONALLY LEFT BLANK
Ex-1015	INTENTIONALLY LEFT BLANK
Ex-1016	INTENTIONALLY LEFT BLANK
Ex-1017	INTENTIONALLY LEFT BLANK
Ex-1018	INTENTIONALLY LEFT BLANK
Ex-1019	INTENTIONALLY LEFT BLANK
Ex-1020	Claim Mapping Table
Ex-1021	INTENTIONALLY LEFT BLANK
Ex-1022	INTENTIONALLY LEFT BLANK
Ex-1023	U.S. Patent No. 7,263,457 ("White")
Ex-1024	U.S. Patent No. 8,438,416 ("Kocev")
Ex-1025	U.S. Patent No. 8,984,523 ("Vajda")
Ex-1026	U.S. Patent No. 7,779,276 ("Bolan")
Ex-1027	Selected pages of Stefanos Kaxiras et al., *Computer Architecture Techniques for Power-Efficiency* (2008)
Ex-1028	U.S. Published Patent Application No. 2010/0162006 ("Therien")

II. LEGAL STANDARDS

9. In forming my opinions and considering the subject matter of the '833 Patent and its claim in light of the prior art, I am relying on certain legal principles that counsel in this case explained to me. My understanding of these concepts is summarized below.

10. I understand that earlier publications and patents may act to render a patent unpatentable for one of two reasons: (1) anticipation, and (2) obviousness.

A. Anticipation

11. It is my understanding that the claims of a patent are anticipated by a prior art reference if each and every element of the claim is found either explicitly or inherently in the reference. I understand that inherency requires a showing that the missing descriptive matter in the claim is necessarily present in the allegedly anticipating reference, and that it would have been so recognized by a person of ordinary skill in the art ("POSITA").

12. I understand that when a challenged claim covers several structures, either generically or as alternatives, the claim is deemed anticipated if any of the structures within the scope of the claim is found in the prior art reference.

13. Although anticipation typically involves the analysis of a single prior art reference, I understand that additional references may be used to show that the prior art reference has enabling disclosure (i.e., allows a POSITA to make the invention without undue experimentation), to explain the meaning of a term used in the prior art reference, and/or to show that a characteristic is inherent in the prior art reference.

B. Obviousness

14. I understand that a claim is invalid as obvious if it would have been obvious to a person of ordinary skill in the art at the time the alleged invention was made. This means that even if all of the elements of the claim cannot be found in a single prior art reference that would anticipate the claim, a person of ordinary skill in the art who was aware of the prior art would have been able to come up with the claimed invention. This may be the case, for example, where the missing element represents only an insubstantial different over the prior art or a reconfiguration of a known system. I understand that in an obviousness determination, the person of ordinary skill in the art is presumed to have knowledge of all material prior art.

15. I understand that an obviousness analysis requires an understanding of the scope and content of the prior art, any differences between the alleged invention and the prior art, and the level of ordinary skill in evaluating the pertinent art.

16. I understand that when a product is available, design incentives and other market forces can prompt variations of it, either in the same field or a different one. If a person of ordinary skill in the art can implement a predictable variation, obviousness likely bars its patentability. For the same reason, if a technique has been used to improve one device and a person of ordinary skill in the art would recognize that it would improve similar devices in the same way, using the technique would have been obvious.

17. I understand that whether a prior art reference renders a patent claim unpatentable as obvious is determine from the perspective of a person of ordinary skill in the art at the time of the alleged invention. I have been told that there is not requirement that the prior art contain an express suggestion to combine known elements to achieve the claimed invention, but a suggestion to combine known elements to achieve the claimed invention may come from the prior art, as filtered through the knowledge of one skilled in the art. In addition, I have been told that the inferences and creative steps a person of ordinary skill in the art would employ are relevant to the determination of obviousness.

18. I understand that one may consider, e.g., whether (1) the change was merely the predictable result of using prior art elements according to their known functions, or whether it was the result of true inventiveness; (2) there is some teaching or suggestion in the prior art to make the modification or combination of elements claimed in the patent; (3) the claimed innovation applies a known technique that had been used to improve a similar device or method in a similar way; (4) the claimed invention would have been obvious to try, meaning that the claimed innovation was one of a relatively small number of possible approaches to the problem with a reasonable expectation of success by those skilled in the art; (5) the invention merely substituted one known element for another known element in order to obtain predictable results; (6) the invention merely applies a known technique to

a known device, method, or product to yield predictable results; or (7) known work in one field of endeavor may have prompted variations of it for use in either the same field or a different one based on design incentives or other market forces that would have been predictable to a person of ordinary skill in the art.

19. I further understand that certain factors may support or rebut the obviousness of a claim. I understand that such secondary considerations include, among other things, commercial success of the patented invention, skepticism of those having ordinary skill in the art at the time of the invention, unexpected results of the invention, any long-felt but unsolved need in the art that was satisfied by the alleged invention, the failure of others to make the alleged invention, praise of the alleged invention by those having ordinary skill in the art, and copying of the alleged invention by others in the field. I understand that there must be a nexus—that is, a connection—between any such secondary considerations and the alleged invention. I also understand that contemporaneous and independent invention by others is a secondary consideration tending to show obviousness.

20. I am not aware of any allegations by the named inventor of the ’833 Patent or any assignee of the ’833 Patent that any secondary considerations tend to rebut the obviousness of the ’833 Patent.

21. Additionally, I understand that in considering obviousness, it is important not to use the benefit of hindsight derived from the patent under consideration.

C. Level of Ordinary Skill in the Art

22. When interpreting a patent, I understand that it is important to identify the relevant art pertaining to that patent, as well as the level of ordinary skill in that art at the time of the claimed invention. The "art" is the field of technology to which the patent is related.

23. I am informed and understand that the POSITA is a hypothetical person who is presumed to know the relevant prior art. I understand that the actual inventor's skill is not determinative of the level of ordinary skill. I further understand that factors that may be considered in determining level of skill include: (i) the types of problems encountered in the art; (ii) prior art solutions to those problems; (iii) the rapidity with which innovations are made; (iv) the sophistication of the technology; and (v) the educational level of active workers in the field. I understand that not all such factors may be present in every case, and one or more of them may predominate.

24. I understand that a POSITA is one who is presumed to be aware of all pertinent art, thinks along conventional wisdom in the art, and is a person of ordinary creativity. A POSITA would have had knowledge of circuit boards and

microcircuits, including components thereof, and related technologies as of September 6, 2011.

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

26. Based on my education and experience, I would have, at least, met the criteria for a POSITA in September 6, 2011, and I still exceed it today.

D. Claim Construction

27. I understand that the United States Patent and Trademark Office interprets claim terms in an *inter partes* review proceeding under the same claim construction standard that is used in a United States federal court. I understand that under this standard, the meaning of claim terms is considered from the viewpoint of a POSITA at the time of the alleged invention.

28. I have been informed that claim terms are generally given their ordinary and customary meaning as understood by one of ordinary skill in the art in light of the specification and the prosecution history pertaining to the patent. I understand,

however, that claims terms are generally not limited by the embodiments described in the specification.

29. I understand that in addition to the claims, specification, and prosecution history, other evidence may be considered to ascertain the meaning of claim terms, including textbooks, encyclopedias, articles, and dictionaries. I have been informed that this other evidence is often less significant and less reliable than the claims, specification, and prosecution history.

III. SUMMARY OF GROUNDS

30. I understand that Petitioner challenges the validity of Claims 1-5, 7, and 13-18 of the '833 Patent. In particular, I understand that Petitioner asserts the following grounds for challenging the claims of the '833 Patent:

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

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

IV. THE '833 Patent

A. Overview of the '833 Patent

31. The '833 Patent is directed to "a power budget of a processor including multiple domains [that] can be dynamically apportioned at run time." Ex-1001, 1:56-58. Fig. 1, reproduced below, is "a flow diagram of a high level method of performing power budget allocations between multiple domains in accordance with an embodiment of the present invention." *Id*. at 30-32.

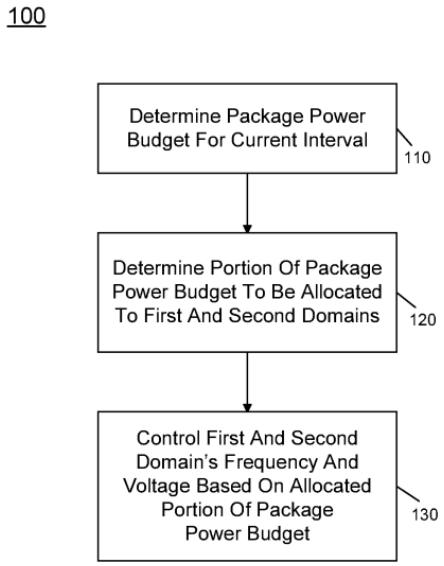


FIG. 1

Ex-1001, Fig. 1.

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

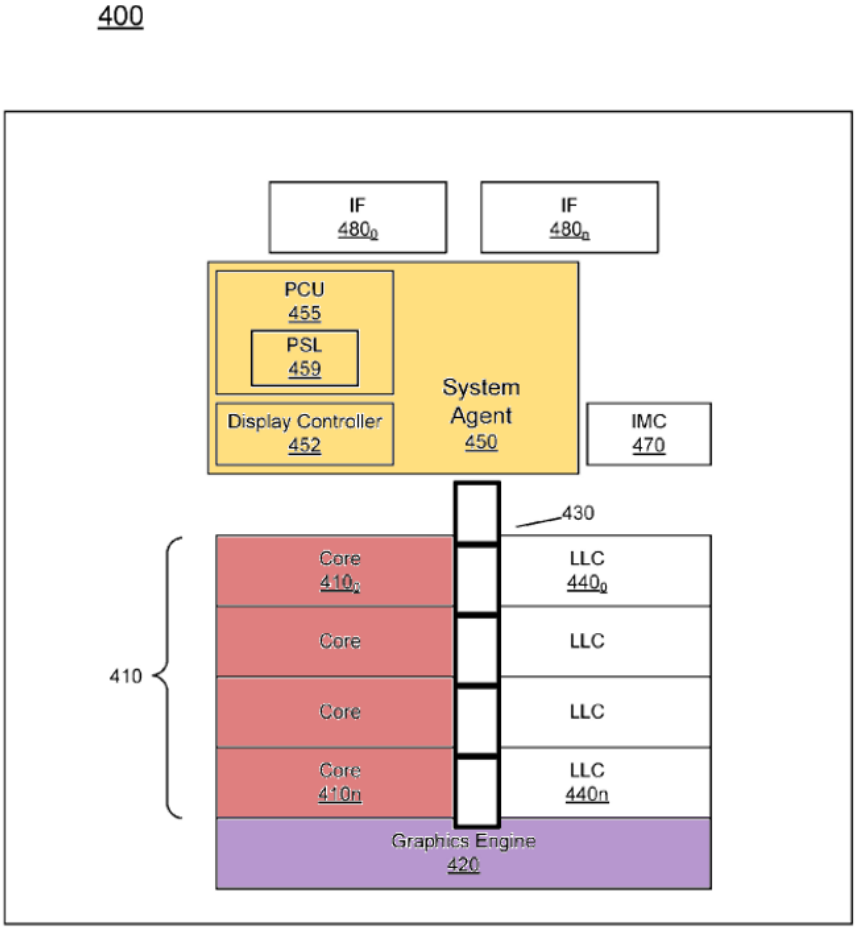


FIG. 7

Ex-1001, Fig. 7 (annotated).

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

to enable the software to determine how package power budget is shared between different domains." *Id*. at 4:1-4.

B. Prosecution History of the '833 Patent

34. I understand the '833 Patent was filed on February 28, 2013, as a continuation of U.S. patent application serial no. 13/225,677, filed on September 6, 2011.

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

36. I understand the prior art relied on in this Petition was not before the PTO during prosecution of the '833 Patent. Thus, I understand that the Petition presents substantially new arguments that were not considered during prosecution.

V. TECHNOLOGY BACKGROUND

37. To provide background for the element-by-element analysis of the claims to follow, below I will present an overview of the state of the art existing at the time of the alleged invention relating to processors generally, and as it relates to multi-core devices, and power, voltage and frequency management approaches, among other relevant technology. As I will describe below from a high-level functional perspective, in the next section based on specific prior art, and then in subsequent sections on a limitation-by-limitation basis, all of these technologies and

techniques were well-known to those of ordinary skill in the art at the time of the alleged invention claimed in the ’833 Patent, and a POSITA would have readily understood the combination of elements of Claim 1-5, 7, and 13-18 to have been obvious.

A. Power, voltage, and frequency control approaches

38. “Limiting power consumption presents a critical issue in computing, particularly in battery powered devices such as laptop computers and cell phones. Limiting power in other computer settings—such as server farms—warehouse-sized buildings filled with Internet service providers’ servers—is also important.”

39. I wrote this in 2000/1 in two review articles, at which time the problem of power consumption was well known and starting to become a design constraint for computer designers. The source of power consumption in integrated circuits is dictated by the voltage that they require and the frequency at which they operate. Power consumption increases in proportion to the frequency at which the core, an integrated circuit, operates but increases as the square of the voltage supply.

40. Additionally, increasing frequency improves performance but requires that the voltage also be increased. Thus, to improve performance by increasing frequency requires a quadratic increase in voltage supply. This in turn has a translates to a significant increase in power consumption.

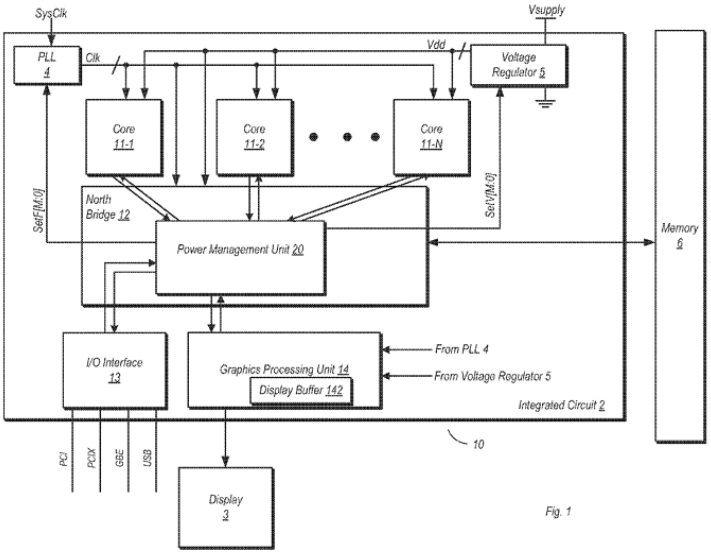
B. Multi-core processing approaches

41. Manufacturers of processor cores, before about the turn of the century, had been able to increase performance by simply increasing frequency. Unfortunately, the fact that power increases as square of the voltage finally caught up with them and meant that increasing frequency quickly became impractical. The solution that began to emerge was to employ multiple processors to do the work of one. The reasoning was that two processors could run at half the frequency, but now the quadratic dependency on supply voltage gave a positive effect, because the voltage supply could be lowered at a quadratic rate. Thus the power consumption of two processors could actually be less than that of one fast processor.

42. This observation led the industry to adopt the idea of multiple processors operating at a lower frequency to replace one fast processor. In about 2005, Intel and its rivals began to perform a "right turn", to quote their promotional material, which meant they moved away from using a single fast processor and replaced it with multiple slower cores on an integrated circuit to meet power limitations, while at the same time maintaining performance. The background of the '833 Patent reflects this concept, which discusses a scenario with multiple cores, and was widely employed at the date of the patent. The multiple cores opened up further ways to control power consumption: when not needed some cores could operate at a lower performance stated or could be turned off.

C. NorthBridge – SouthBridge Architecture

43. A NorthBridge is a common component of many processors. The NorthBridge may include "separate interfaces to communicate with and route communication from multiple logic cores of an integrated circuit to other sub-systems, components or peripherals external to the integrated circuit." Ex-1023. U.S. Patent No. 8,438,416, filed October 21, 2010 utilizes a NorthBridge and "is directed to a method and apparatus for function-based dynamic power control of a bridge unit, such as a north bridge unit in a computer system." Ex-1024, 2:44-46. The NorthBridge may also include "an interface to an I/O bridge (e.g., a South bridge), a memory controller, and a graphics processing unit." *Id*. at 2:57-59.



VI. THE PRIOR ART IN THE APPLIED INVALIDITY GROUNDS

A. Finkelstein

44. Finkelstein, titled "Power management for multiple processor cores," published on May 6, 2010. Ex-1005, Cover.

45. I understand Finkelstein is prior art under at least 35 U.S.C. § 102(b).

46. Finkelstein "relates to power management for multiple processor cores." Ex-1005 ¶1. As shown in annotated Figure 1 below, Finkelstein discloses a system that includes "one or more processors" (purple rectangle) where each processor "may include one or more processor cores" (red). *Id.* ¶¶10-11.

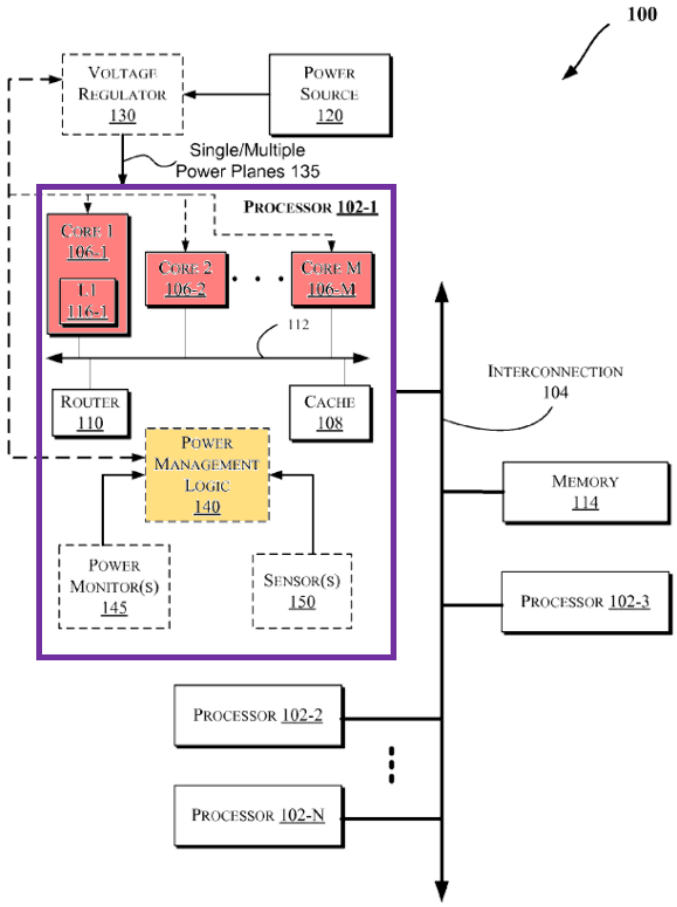


FIG. 1

Ex-1005, Fig. 1 (annotated).

47. Finkelstein explains that each processor may communicate with various other components, including memory, voltage regulator(s), and power source(s). Ex-1005 ¶¶13-14. A voltage regulator "may be coupled to a single power plane … or to multiple power planes … where each power plane may supply power to a different core or group of cores." *Id*. ¶14. Finkelstein's processors may further "include one or more shared and/or private caches…, buses or interconnections…, graphics and/or memory controllers…, or other components." *Id*. ¶11.

48. Finkelstein also discloses "a power management logic 140 [colored yellow in Figure 1, above] to control supply of power to" processor components. Ex-1005 ¶¶16, 24, 28-36. Finkelstein explains that power management logic 140 "may request the cores 106 to modify their operating frequency, power consumption, etc." *Id*. It may also "consider statistics … on operating states of the cores" and "select[] technique(s) [that] may be applied to throttle or modify an operational characteristic of one or more of the cores." *Id*. ¶24. Finkelstein also discloses that "a controller budget may be defined (e.g., implemented by the logic 140)" and provides several formulae by which logic 140 may dynamically allocate the power budget between power planes. *Id*. ¶¶28-36. For example, Finkelstein discloses sharing an "energy budget" so that "unused processor core power" can be used for other cores that may be performing more "intensive workload." *See*, *e.g.*, *id*. ¶26.

Finkelstein further discloses that "user (or application defined) preferences" may be considered in determining power budget allocation. *Id*. ¶34; *see also id*. ¶¶29, 31.

B. Conroy

49. Conroy, titled "Methods and apparatuses for dynamic power control," published on March 1, 2007. Ex-1006, Cover.

50. I understand Conroy is prior art under at least 35 U.S.C. § 102(b).

51. Conroy discloses "dynamically redistribut[ing] power in a [computer] system that includes a plurality of subsystems." Ex-1006, Abstract. For example, Conroy teaches identifying a "load profile of the system" and redistributing power "between the subsystems while tracking the load profile." *Id*. Conroy discloses that its "subsystems may be, for example, a central processing unit ('CPU') and a graphics processor ('GPU')," and that a "power weighting arrangement between the subsystems may be a power distribution table selected out of a plurality of power distribution tables" that describe, among other things, "the power used by each of the subsystems at various system's settings." *Id*. ¶¶5-6. Examples of power distribution tables are illustrated in Figures 21A-C and 22, copied below.

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

2100

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

2105

Fig. 21 A

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

2110

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

2105

Fig. 21 B

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

2120

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

2105

Fig. 21 C

Load Profile Kn

2200

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

2206

Fig. 22

Ex-1006 Figs. 21A-21C, 22.

52. Conroy's power distribution table may correspond to various power distribution profiles, including where one subsystem requires most of the power followed by another system requiring most of the power. For example, Conroy explains that during a "CPU-heavy load profile," when CPU workload is intensive, power may be distributed such that "CPU consumes about 75%, while GPU

consumes about 25% of the power allocated for the system." Ex-1006 ¶203. And, if another system, such as the GPU, is engaged in intensive workload after the CPU, "the power of the system is shifted towards GPU" such that, for example, "GPU consumes about 75%, while CPU consumes about 25% of the power allocated for the system." *Id*.

C. Nussbaum

53. Nussbaum, titled "Determining performance sensitivities of computational units," published on January 27, 2011 from an application filed on July 24, 2009.

54. I understand Nussbaum is prior art under at least 35 U.S.C. §§ 102(a) and 102(e).

55. Nussbaum discloses power management of computational units in a "System on a Chip (SOC)" and allocation of "power headroom" to various "on-die or on-platform components." Ex-1007, Abstract, ¶20. As shown in annotated Figure 1, Nussbaum's SOC "includes multiple CPU processing cores 101" (red) operating in a first domain. *Id*. ¶20. Nussbaum's SOC also includes "GPU (Graphics Processing Unit) 103" (purple) operating in a second domain. *Id*. Furthermore, Nussbaum's SOC includes "North-Bridge & Memory Controller 107" (green) operating in a third domain, which may include "performance analysis control logic 111" and "power allocation controller 109" (yellow). *Id*. Nussbaum

explains that "performance analysis control logic 111" and "power allocation controller 109" (yellow) are used to "analyze[] performance sensitivity of the cores and other computational units" and "control[] allocation of the Thermal Design Point (TDP) power headroom to the on-die or on-platform components." *Id.* ¶¶20, 39.

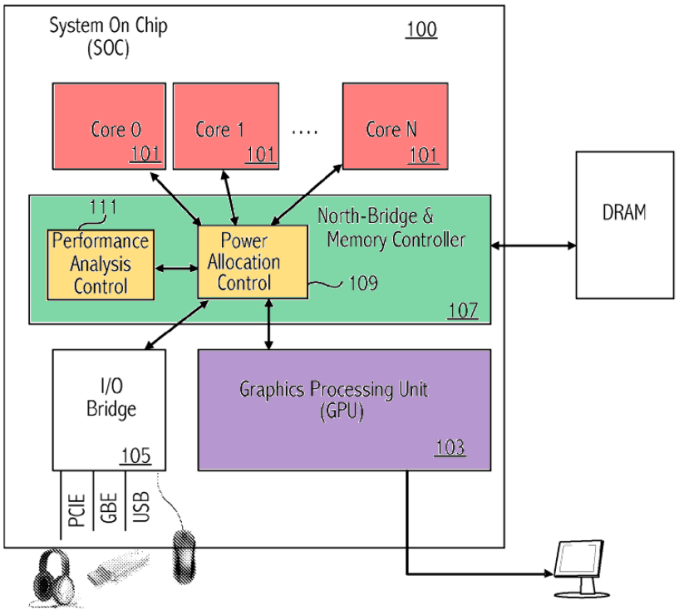


FIG. 1

Ex-1007, Fig. 1 (annotated).

56. Nussbaum also discloses "reallocation of the power from idle or less active components to the busy components by having more power allocated to the busy ones." Ex-1007 ¶23. Nussbaum explains that, for example, "one way to determine how to allocate power between Core0 and Core1 … is to know which of the two cores, if any, can better exploit an increase in performance capability provided, e.g., by an increase in frequency." *Id.* ¶25. Nussbaum also explains that "performance sensitivity of each computational unit to frequency change, and/or

other change in performance capability, also referred to herein as boost sensitivity, is determined and stored on a computational unit basis." *Id*.

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

D. Bose

58. Bose, titled "Method and system of peak power enforcement via autonomous token-based control and management," published on April 2, 2009 from an application filed on September 27, 2007.

59. I understand Bose is prior art under at least 35 U.S.C. § 102(b).

60. Bose discloses "power management of a system of connected components … wherein each component is assigned a power budget as determined by a number of allocated tokens in the token allocation map." Ex-1008, Abstract. Bose utilizes a "token management unit (TMU)" to manage allocation of tokens to "token-managed units," such as "processor resources," and explains that "[e]ach

managed component, in response to its own operating conditions (as evidenced by its utilization markers) can either attempt to donate one or more token(s)" or "read (receive) one or more token(s)." *Id*. ¶¶38-39, 68. "Each token connotes a pre-specified quantum of power, and as such, the total number of tokens owned by a component determines its own power limit." *Id*. ¶68.

61. Bose also describes an implementation in which the "TMU fixes a budget of 3 tokens for all other non-managed units (including itself), and distributes the remaining 12 tokens across the 4 managed units." Ex-1008 ¶53. Bose further discloses that if the performance level of a particular resource is "below a predefined minimum performance," a process is initiated to "try and read additional tokens from [a] centralized bus token storage … to boost up performance" of that particular resource, e.g., "by increasing its voltage and frequency, if allowed by the system architecture." *Id*. ¶86, Fig. 13. Bose explains that if no additional tokens can be allocated to the particular resource at the moment, that particular resource can wait for its token count to be increased in the next iteration of the process. *Id*.

VII. CLAIM CONSTRUCTION

62. I do not believe that any claim term requires explicit construction to resolve the issues presented in this Petition. I apply the plain and ordinary meaning to each claim term, as that plain and ordinary meaning would have been understood by a POSITA.

63. 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.

VIII. DETAILED EXPLANATION OF THE UNPATENTABILITY GROUNDS

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

1. Independent claim 1

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

64. In my opinion, Finkelstein teaches the preamble of Claim 1, to the extent it is limiting.

65. Finkelstein discloses "processors" that "have corresponding voltage regulator(s) and/or power source(s)" as well as "multiple power planes 135 (e.g., where each power plane may supply power to a different core or group of cores)." Ex-1005 ¶14; *see also id*. ¶¶11, 16, 24. An example of Finkelstein's processor 102-1 (purple) is shown in annotated Figure 1 below.

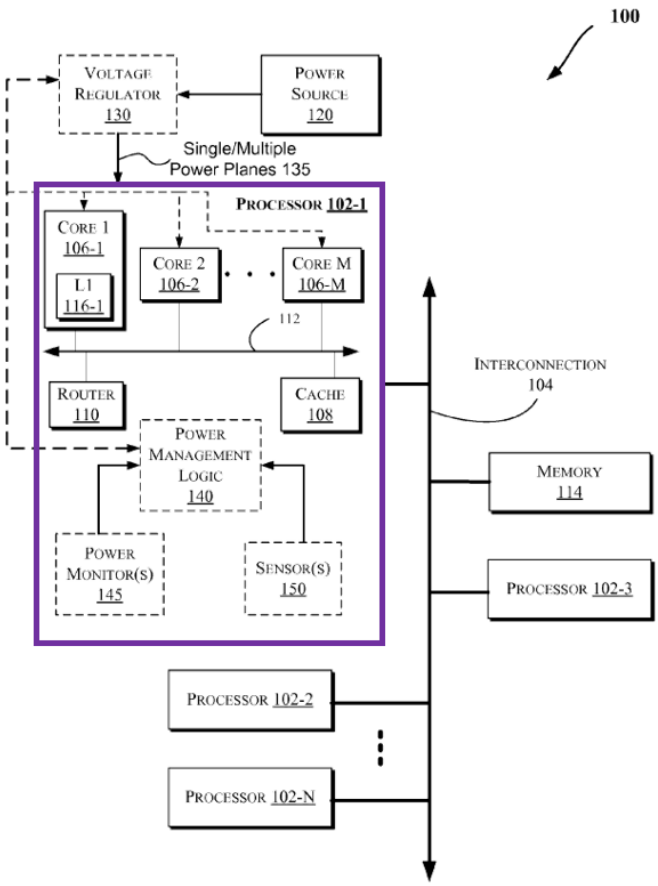


FIG. 1

Ex-1005, Fig. 1 (annotated).

66. Thus, it is my opinion that Finkelstein discloses a processor as described in the preamble.

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

67. In my opinion, Finkelstein teaches Element 1[A].

68. As shown in annotated Figure 1 below, Finkelstein discloses cores in "multiple power planes 135 (e.g., where each power plane may supply power to a different core or group of cores)." Ex-1005 ¶14. A POSITA would have understood

that each "power plane" 135 corresponds to a "domain." Indeed, the '833 Patent itself refers to "domains" as "planes." Ex-1001, 4:7-21. Thus, a POSITA would have understood that a first set of processor cores, e.g., Core 1 (red) in a first power plane, is a "first domain" and another set of cores, e.g., Core 2 (blue) in a second power plane, is a "second domain."

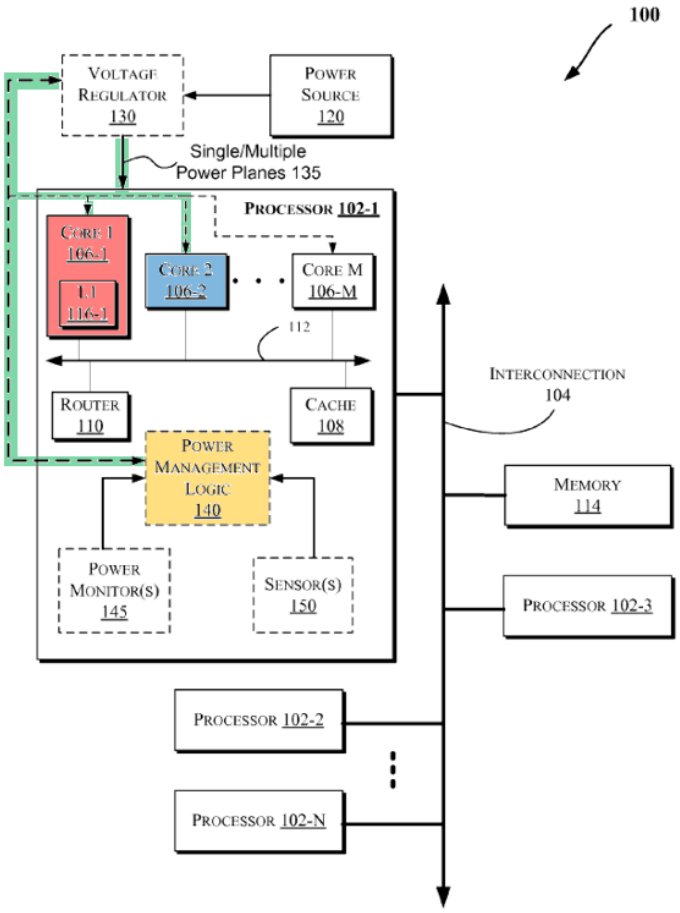


FIG. 1

Ex-1005, Fig. 1 (annotated).

69. Finkelstein further discloses that each of the first and second domains operates at an independent voltage and frequency. Finkelstein uses "power

management logic 140 to control supply of power to components of the processor 102 (e.g., cores 106)," and power management logic 140 (above, yellow) "may request the cores [] to modify their operating frequency, power consumption, etc." Ex-1005 ¶16; *see also id*. ¶31. Logic 140 can also "throttle or modify an operational characteristic of ***one*** or more of the cores 106 (such as an operating voltage and/or an operating frequency of ***a processor core*** 106)." *Id*. ¶24.[2] A POSITA implementing Finkelstein would thus have been motivated to operate Core 1 (in a first domain) and Core 2 (in a second domain) at independent voltages and frequencies by controlling the "operational characteristic of one" of the cores such that logic 140 could modify the operating voltage and/or frequency of either Core 1 or Core 2, without affecting the operating voltage and/or frequency of the other. This independent operation would have allowed for a greater degree of control over one core, while still allowing for increased performance on the other. For example, one core that was operating too hot could be set to a lower frequency, while the other core could continue to operate at a higher frequency and performance. *Id*.; *see also id*. ¶¶27, 32, 39 (discussing control of power budget "per power plane"); Ex-1027 at 35-40 (describing "independent" voltage and frequency control for each core or domain).

[2] Throughout this Declaration, all annotations and emphasis have been added, unless otherwise noted.

70. Finkelstein further discloses mechanisms that facilitate independent control of voltage and frequency to different domains. For example, as shown in annotated Figure 1 above, connections (green) are established between logic 140 (yellow) and Cores 1 and 2 (red and blue) through voltage regulator 130, enabling logic 140 to "request the cores [] to modify their operating frequency, power consumption, etc." Ex-1005 ¶16; *see also id*. ¶31. A POSITA would have understood that Finkelstein teaches operating the first and second domains at an independent voltage and frequency because it discloses fulfilling different voltage and frequency needs for different domains (e.g., the first domain containing Core 1 and the second domain containing Core 2) and discloses mechanisms that facilitate independent control of voltage and frequency to different domains.

71. Thus, it is my opinion that Finkelstein teaches a first domain and a second domain, each of the first and second domains to operate at an independent voltage and frequency.

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

72. In my opinion, Finkelstein teaches Element 1[B].

73. Finkelstein discloses that processor cores "may be implemented on a single integrated circuit (IC) chip," which in turn may include "memory controllers (such as those discussed with reference to FIGS. 6-7)." Ex-1005 ¶11. "Moreover,

various components of the processor 102-1 may communicate with the cache 108 directly, through a bus (e.g., the bus 112), ***and/or a memory controller or hub***." *Id*. ¶13. It was well known that in implementations where a memory controller is used to facilitate communications between various components of processor 102-1 (including cores 106) and cache 108, each core 106 would be couple to the memory controller so that each core 106 can communicate with cache 108 through the memory controller.

74. An example of this is shown in annotated Figure 7 (referred to in the description of Figure 1), which depicts processors 702 and 704, each including "a local memory controller hub (MCH)" (green). *Id*. ¶49. Finkelstein also explains that processors 702 and 704, each including "a local memory controller hub (MCH)," may include "one or more of the cores 106, logic 140, sensor(s) 150, and/or power monitor(s) 145 of FIG. 1." *Id*. ¶49.

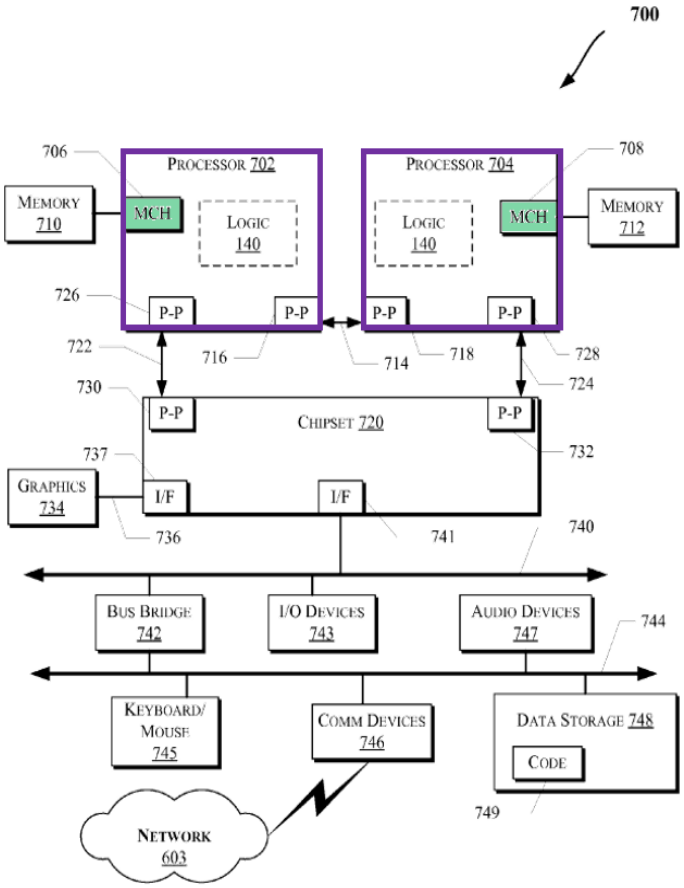


FIG. 7

Ex-1005 Fig. 7 (annotated).

75. Thus, it is my opinion that Finkelstein teaches a memory controller coupled to the first domain (e.g., Core 1) and the second domain (e.g., Core 2).

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

76. In my opinion, Finkelstein teaches Element 1[C].

77. As illustrated in annotated Figure 1 below, Finkelstein discloses that processor cores "may be implemented on a single integrated circuit (IC) chip," which in turn may include "buses or interconnections (such as a bus or

interconnection 112)." Ex-1005 ¶11. Finkelstein also explains that "various components of the processor 102-1" may communicate with cache 108 and router 110 through interconnection 112 (green). *Id*. ¶¶11-13, Fig. 1. A POSITA would have understood Finkelstein as teaching an "interface" because an "interconnection" for communications between various components of processor 102-1, cache 108, and router 110 serves as an "interface" between these components. *Id*. ¶¶10-13, 44-52. Indeed, Finkelstein itself discloses using "interfaces" to "interconnect[]" processors, memory, and input/output devices, and a POSITA would have understood to implement interconnection 112 (green) using the same or similar "interfaces." *Id*. ¶¶47-48. The '833 Patent also explains that an "interface" enables "interconnection" between the processor cores and other circuitry. *See, e.g.,* Ex-1001, 9:58-60 ("Multiple interfaces … may be present to enable interconnection between the processor and other circuitry.").

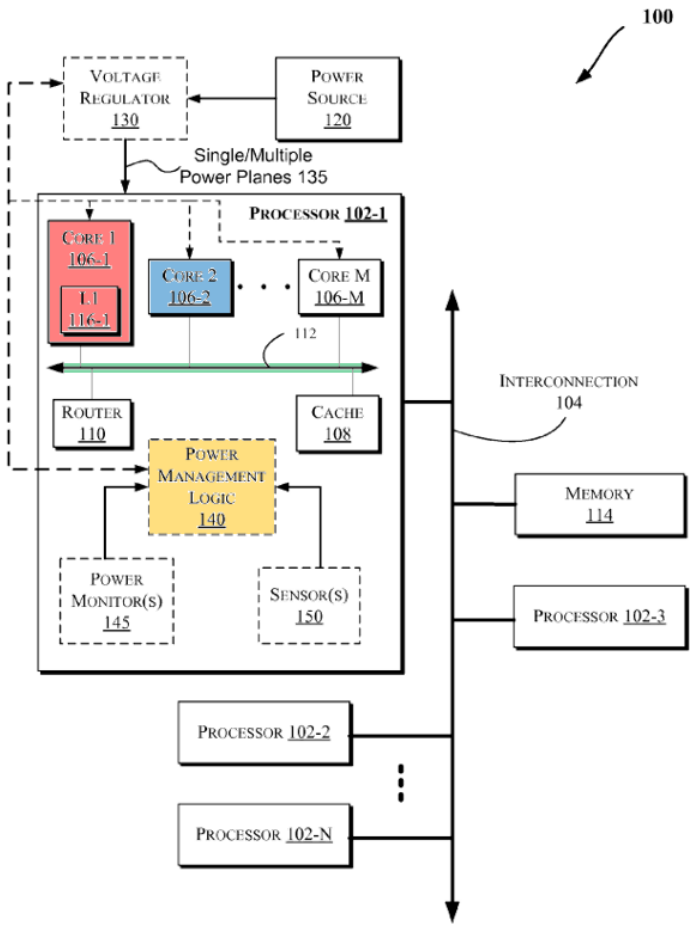


FIG. 1

Ex-1005 Fig. 1 (annotated).

78. Thus, it is my opinion that Finkelstein teaches at least one interface.

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

79. In my opinion, Finkelstein teaches Element 1[D.i].

80. Finkelstein discloses that "if multiple power planes [are] sharing the same power source," then these planes may "share a common package power/energy budget" so that, for example, "unused processor core power" of a core in the first

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

81. Thus, it is my opinion that Finkelstein teaches a first logic to dynamically allocate a power budget for the processor between the first and second domains at run time.

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

82. In my opinion, Finkelstein teaches Element 1[D.ii].

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

(1) First Approach

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

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

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

value for the second domain (receiving power from power plane i=2). Furthermore, Finkelstein explains that "unused" budget for a particular power plane "may be distributed among the rest of the power planes proportionally to" these sharing policy values, i.e., W_{k1} and W_{k2} (referred to as weights), and that these sharing policy values are controllable by user-level software because they are "***user*** [] preferences" and a POSITA would have understood that such preferences are provided by users via user-level software (e.g., via a graphical interface). *Id*. ¶¶29, 31, 34; *see also* Ex-1028 ¶17 ("The preference might be determined … based on a selection by a user of the system (e.g., via a graphical interface).").

(2) Second Approach

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

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

Ex-1005 ¶28. In equation (1), "TDP_n is the Thermal Design Power (TDP) power limit on step n," "P_n is the power spent on step n over time Δt_n," "α is the decay component," and "En corresponds to the energy 'remainder' which is the amount of

energy not consumed by the system." *Id*. ¶29. Finkelstein explains that "the system (e.g., logic 140) may set ***TDP constraints on each of the power planes separately***," and notes that "this framework smoothly handles a case when ***TDP is changed on-the-fly (e.g., by a software application or a user)***." *Id*. ¶31. Thus, in equation (1), TDP constraints on a first power plane (the power plane that supplies power to the first domain) represents a first sharing policy value and TDP constraints on a second power plane (the power plane that supplies power to the second domain) represents a second sharing policy value. Furthermore, these sharing policy values, i.e., TDP constraints on each of the power planes, are controllable by user-level software because they can be "changed on-the-fly" by a user and a POSITA would have understood that such changes are made via user-level software (e.g., via a graphical interface). *Id*.; *see also* Ex-1028 ¶17 (using a graphical interface to receive input from a user).

86. Thus, it is my opinion that Finkelstein discloses at least two approaches where the first logic may dynamically allocate a power budget for the processor between the first and second domains at run time according to a first sharing policy value for the first domain and a second sharing policy value for the second domain, the first and second sharing policy values controllable by user-level software.

2. Dependent claims 2-5 and 7

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

87. In my opinion, Finkelstein teaches Claim 2.

88. As explained above in Sections VIII.A.1.e-f, Finkelstein discloses that the first logic determines a portion of the power budget to allocate to the first domain based on the first sharing policy value. Moreover, as described above with respect to the "First Approach," logic 140 implements power distribution, according to vector W_{ki}, wherein "entry i corresponds to the ***portion of the budget that goes to power plane i***":

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

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

89. Thus, it is my opinion that Finkelstein discloses that the first logic is to determine a portion of the power budget to allocate to the first domain based on the first sharing policy value.

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

90. In my opinion, Finkelstein teaches Claim 3.

91. As explained above in Sections VIII.A.1.e-f, Finkelstein's processors include logic 140 that dynamically allocates the power budget at run time. Finkelstein also discloses that "a controller budget may be defined (e.g., implemented by the logic 140)" and provides several equations by which logic 140 may dynamically allocate the power budget.

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

$$f^i(0)=P_1^{\,1}$$

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

Ex-1005 ¶32; *see also id*. ¶¶28, 30, 34, 36.

92. Finkelstein explains that $f^i(E)$ is a "controller function" defined "[p]er power plane i" and is used to "map[] energy budget onto [a] discrete set of power states." Ex-1005 ¶32. As shown in annotated Figure 4 below, $f^i(E)$ "maps a range [E_{low}^i, E_{high}^i] onto [a] discrete range of [P_n^i, P_0^i]," where P_n^i is the "maximal efficiency state," which a POSITA would have understood to mean the most power efficient, i.e., the lowest power consumption state. *Id*. Finkelstein also explains that

"[b]udget values below E_{low}^{i} are mapped into P_n^i," hence the range [P_n^i, P_0^i], confirming that P_n^i represents the minimum reservation value for power plane i because there is no power state below P_n^i. *Id*. Finkelstein therefore discloses a first minimum reservation value P_n^1 for the first power plane and a second minimum reservation value P_n^2 for the second power plane.

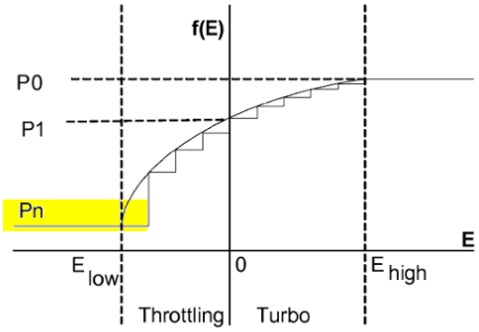


FIG. 4

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

power planes to satisfy P_n^i for each power plane. *Id.*; *see also* Ex-1026, 3:65-67, 7:18-22, 11:13-20 (describing a "user preferences database" containing "parameters and procedures for how to apportion power among various devices," including "trigger conditions" such as when the power margin for one or more devices drops below a "selected minimum power margin"); Ex-1028 ¶17 ("The preference might be determined … based on a selection by a user of the system (e.g., via a graphical interface).").

94. Thus, it is my opinion that Finkelstein discloses that the first logic is to dynamically allocate the power budget further according to a first minimum reservation value for the first domain and a second minimum reservation value for the second domain, the first and second minimum reservation values controllable by the user-level software.

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

95. In my opinion, Finkelstein teaches Claim 4.

96. As explained above for Claim 3, Finkelstein discloses a first logic that provides at least a first portion of the power budget to the first domain, the first portion corresponding to the first minimum reservation value. Specifically,

Finkelstein discloses providing a first minimum reservation value, P_n^1, to the first power plane. Ex-1005 ¶32.

97. Thus, it is my opinion that Finkelstein discloses that the first logic is to provide at least a first portion of the power budget to the first domain, the first portion corresponding to the first minimum reservation value.

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

98. In my opinion, Finkelstein teaches Claim 5.

99. As explained in Sections VIII.A.1.e-f, specifically the "Second Approach," Finkelstein discloses that the first logic (logic 140) is to determine the power budget for a current time interval based at least in part on a power budget carried forward from a previous time interval, a power consumed in the previous time interval, and a power budget decay value. Finkelstein explains that "energy budget may be managed and/or power setting(s) (e.g., voltage and/or frequency changes) may be made ***accordingly to a current budget***," and that "[e]nergy-based power management may be performed by controlling the energy budget defined iteratively as:"

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

where "TDP_n is the Thermal Design Power (TDP) power limit on step n," "P_n is the power spent on step n over time Δt_n," "α is the decay component," and "E_n corresponds to the energy 'remainder' which is the amount of energy not consumed by the system." Ex-1005 ¶¶28-29.

100. A POSITA would have understood that E_n corresponds to a power budget carried forward from a previous time interval because E_n is "the energy 'remainder' which is the amount of energy not consumed by the system."[3] Ex 1005 ¶¶28-29. Similarly, a POSITA would have understood that P_n corresponds to a power consumed in the previous time interval because P_n is "power spent on step n over time Δt_n" and that α corresponds to a power budget decay value because α is "the decay component." *Id*. Furthermore, as shown in Figure 5 below, Finkelstein discloses power management operations in which a new cycle begins with "determin[ing] energy budget remainder," then "determin[ing] recommendations per power plane." *Id*. ¶39. Hence, the power budget allocation in the current cycle is determined, at least in part, on the power budget carried forward from the last cycle.

[3] The '833 Patent refers to E_n as both a "power budget" (*see* Claim 6), measured in watts, and as an "energy budget," measured in joules, Ex-1001, 3:27-33. A POSITA would recognize that energy is the time integral of power (e.g., a joule is a watt-second). Therefore, E, whether it is called a power budget over an interval, Δt, as in Claims 5 and 6, or an energy budget over the same interval, the two things refer to the same quantity.

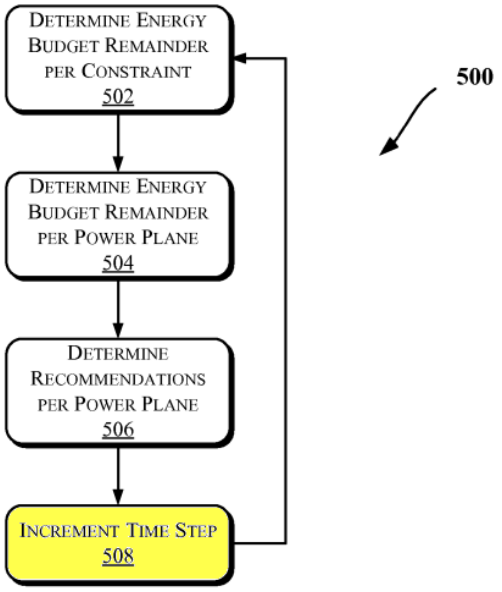


FIG. 5

Ex-1005 Fig. 5 (annotated).

101. Thus, it is my opinion that Finkelstein discloses that the first logic is to determine the power budget for a current time interval based at least in part on a power budget carried forward from a previous time interval, a power consumed in the previous time interval, and a power budget decay value.

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

102. In my opinion, Finkelstein teaches Claim 7.

103. Finkelstein discloses that "if multiple power planes sharing the same power source," then these planes may "share a common package power/energy budget" so that, for example, "unused processor core power" of a core in one domain

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

104. Moreover, a POSITA seeking to implement the teachings of Finkelstein would have found it obvious that Finkelstein's power management logic 140 would dynamically allocate substantially all of the power budget to the second domain for a second workload executed after the first workload. A POSITA would have understood that processors encounter both graphic intensive workload and data processing intensive workload. *See*, *e.g.*, Ex-1006 ¶196 (recognizing that many applications "present an alternating high workload to the CPU and to the GPU"); Ex-1007 ¶22 ("Most real-world applications are either CPU or GPU-bounded."). A POSITA would have also understood that processors may execute data processing

intensive workload after execution of graphic intensive workload, and vice versa. *Id*. A POSITA would therefore have understood that after the core(s) in a first domain finishes performing "intensive workload" in GFX, another core in another domain may be re-allocated substantially all of the power budget (previously allocated to the first domain to perform "intensive workload" in GFX) to perform data processing intensive workload. *See*, *e.g.*, Ex-1006 ¶196 (allocating a substantially bigger share of the power to CPU or GPU based on whether a load profile is CPU or GPU-heavy); Ex-1007 ¶¶22, 44-49 (allocating a power headroom to CPU or GPU based on whether an application is CPU or GPU-bounded). Indeed, this would have been an obvious implementation step because Finkelstein already teaches to provide "***the entire budget***" to one power plane (domain). It thus would have been obvious to provide "***the entire budget***" to either domain, for essentially the same reasons and implemented in essentially the same way as taught by Finkelstein.

105. Thus, it is my opinion that Finkelstein discloses that the first logic is to dynamically allocate substantially all of the power budget to the first domain for a first workload, and to dynamically allocate substantially all of the power budget to the second domain for a second workload executed after the first workload, notwithstanding that the lower-bound of "substantially all" may not be reasonably ascertainable in other contexts.

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

1. Dependent claim 7

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

106. In my opinion, Finkelstein in view of Conroy teaches Claim 7.

107. As discussed in Section VIII.A.2.e, Finkelstein teaches a first logic that dynamically allocates substantially all of the power budget to the first domain for a first workload, and dynamically allocates substantially all of the power budget to the second domain for a second workload executed after the first workload. Moreover, a POSITA seeking to implement the power management logic of Finkelstein would also have found this feature obvious in view of Conroy.

108. Like Finkelstein, Conroy relates to "[m]ethods and apparatuses for dynamic power control." Ex-1006, Title. Conroy also discloses a "power weighting arrangement between the subsystems," and explains that "higher power draw is allocated to and allowed for the subsystem having a higher workload." *Id*. ¶6. Conroy further recognizes that many applications "present an ***alternating*** high workload to the CPU and to the GPU that results in an alternating asymmetric load profile of the system." *Id*. ¶196.

109. A POSITA would have understood that Finkelstein's processors may also execute applications that "present an alternating high workload to the CPU and to the GPU that results in an alternating asymmetric load profile of the system." Ex-1006 ¶196. Indeed, many applications present an alternating high workload to the CPU and to the GPU during execution. *See*, *e.g.*, *id*. ¶196; Ex-1007 ¶22 ("Most real-world applications are either CPU or GPU-bounded."). Thus, a POSITA would have been motivated to combine the teachings of Conroy with the teachings of Finkelstein and use Finkelstein's processors to execute a "CPU-heavy load profile" after execution of a "GPU-heavy load profile," and vice versa.

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

111. Thus, it is my opinion that Finkelstein in view of Conroy teaches that the first logic is to dynamically allocate substantially all of the power budget to the first domain for a first workload, and to dynamically allocate substantially all of the power budget to the second domain for a second workload executed after the first workload, notwithstanding that the lower-bound of "substantially all" may not be reasonably ascertainable in other contexts.

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

1. Independent claim 13

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

112. In my opinion, Nussbaum teaches the preamble of Claim 13, to the extent it is limiting.

113. As shown in Figure 1, Nussbaum discloses "System on a Chip (SOC) system" 100. Ex-1007 ¶¶10, 31, 43.

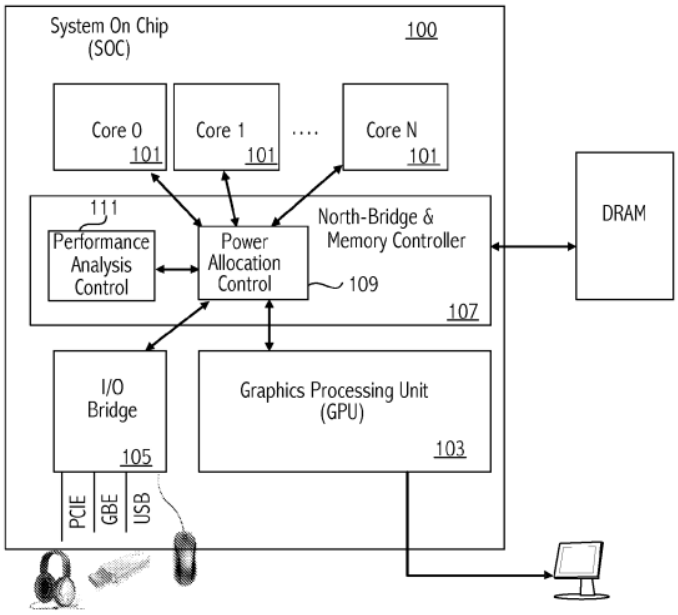


FIG. 1

Ex-1007 Fig. 1.

114. Thus, it is my opinion that Nussbaum discloses a system as described in the preamble.

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

115. In my opinion, Nussbaum teaches Element 13[A.i].

116. As shown in annotated Figure 1 below, Nussbaum's SOC 100 "includes multiple CPU processing cores 101." Ex-1007 ¶20.

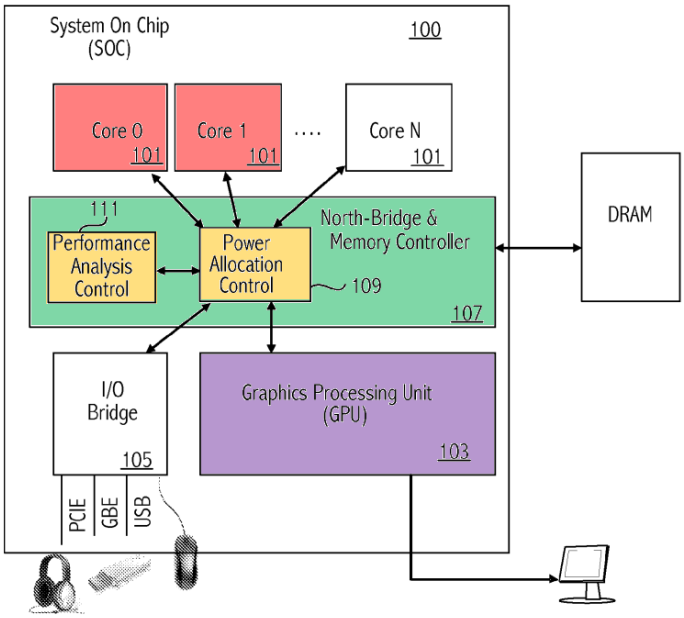


FIG. 1

Ex-1007 Fig. 1 (annotated).

117. Nussbaum also provides examples of power allocation in Tables 1 and 2, copied below, where a plurality of cores 101 (Core 0 and Core 1, red) are equally boosted to a "higher F, V to fill new power headroom." Ex-1007 ¶¶21-24.

TABLE 1

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

TABLE 2

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

118. A POSITA would have understood this to mean that Core 0 and Core 1 are operating in the same power domain. This understanding is consistent with the '833 Patent's description of a domain, because a POSITA would have recognized that Nussbaum's Core 0 and Core 1 form "a collection of hardware and/or logic that operates at the same voltage and frequency point." Ex-1001, 1:58-60. Moreover, consistent with power allocations depicted in Tables 1 and 2 above, a POSITA seeking to implement the teachings of Nussbaum would also have found it obvious that multiple cores may operate in the same power domain (e.g., Core 0 and Core 1 may operate in a first domain while Core 2 and Core 3 may operate in another domain). *See*, *e.g.*, Ex-1005 ¶14 (explaining that each power plane (domain) may supply power to a "group of cores").

119. Furthermore, as shown in annotated Figure 1 and Tables 1 and 2 above, SOC 100 includes "GPU (Graphics Processing Unit) 103" (purple), which operates

at different wattages compared to Core 0 and Core 1 (red). Ex-1007 ¶¶20-24. A POSITA would have understood this to mean that the GPU is not operating in the same domain as Core 0 and Core 1. In other words, the GPU is operating in a second domain. *Id*.

120. Thus, it is my opinion that Nussbaum discloses a multicore processor having a first domain including a plurality of cores and a second domain including a graphics engine.

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

121. In my opinion, Nussbaum teaches Element 13[A.ii(1)].

122. As shown in annotated Figure 1, SOC 100 includes "North-Bridge & Memory Controller 107" (green), which includes "performance analysis control logic 111" and "power allocation controller 109" (yellow). Ex-1007 ¶20. Nussbaum teaches that "performance analysis control logic 111" and "power allocation controller 109" jointly serve as a system agent to "analyze[] performance sensitivity of the cores and other computational units" and "control[] allocation of the Thermal Design Point (TDP) power headroom to the on-die or on-platform components." *Id*. ¶¶20, 39. Nussbaum therefore discloses a "system agent" similar to that described in the '833 Patent, which explains that a "system agent" is a component that controls power supply to processor components. *See*, *e.g.*, Ex-1001, 9:16-21 ("[S]ystem

agent domain 450 may execute at a fixed frequency and may remain powered on at all times to handle power control events and power management such that domains 410 and 420 can be controlled to dynamically enter into and exit low power states."), 9:48-50 ("[S]ystem agent domain 450 may include a power control unit 455 which can include a power sharing logic 459 in accordance with an embodiment of the present invention.").

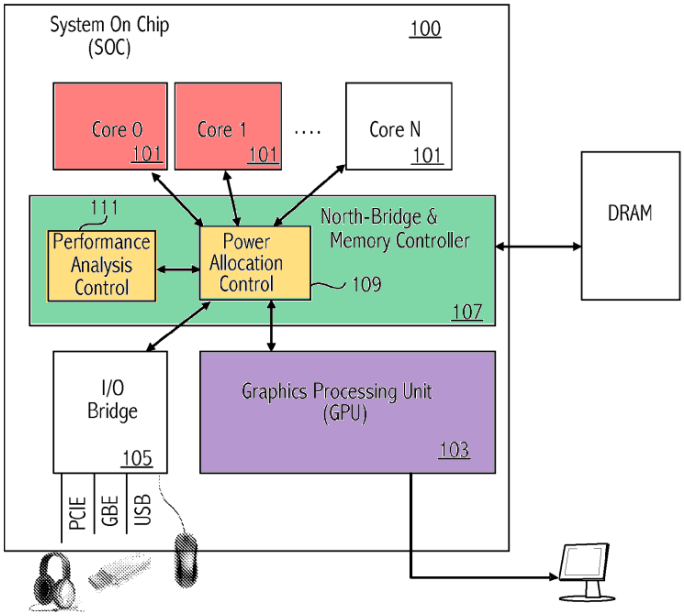


FIG. 1

Ex-1007 Fig. 1 (annotated).

123. Nussbaum also discloses that the system agent circuitry (performance analysis control logic 111 and power allocation controller 109, yellow) is included in North-Bridge & Memory Controller 107 (green) that operates in a third domain. Nussbaum provides an example of power reallocation in Tables 1 and 2, copied below, in which the power allocated to "Memory Controller" remains fixed at "2 w"

despite changes made to other components (including Core 0 and Core 1 in the first domain and GPU in the second domain). Ex-1007 ¶¶21-24.

TABLE 1

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

TABLE 2

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

124. Accordingly, a POSITA would have recognized that, because "Memory Controller" is combined with North-Bridge 107 and not called out separately in Tables 1 and 2, Nussbaum teaches that North-Bridge & Memory Controller 107 operates in a domain different from that of Core 0 and Core 1 (the first domain) and GPU (the second domain). In other words, North-Bridge & Memory Controller 107, which includes the system agent circuitry, is operating in a third domain.

125. Thus, it is my opinion that Nussbaum discloses a multicore processor having a third domain including system agent circuitry.

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

126. In my opinion, Nussbaum teaches Element 13[A.ii(2)].

127. As explained for Element 13[A.ii(2)] above, Nussbaum discloses "reallocation of the power from idle or less active components to the busy components by having more power allocated to the busy ones." Ex-1007 ¶23. Nussbaum provides an example of this reallocation of power in Tables 1 and 2, copied below, where Table 2 depicts "a workload sample where 2 out of 4 cores are idle and GPU operates at half power." *Id*. Notably, as shown in Tables 1 and 2, power allocated to "Memory Controller" remains fixed at "2 w" despite changes made to other components.

TABLE 1

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

TABLE 2

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

128. Accordingly, a POSITA seeking to implement the teachings of Nussbaum would have found it obvious to operate North-Bridge & Memory Controller 107 at a fixed power budget so that the power allocated to North-Bridge & Memory Controller 107 can remain fixed despite changes made to other components. *See* Ex-1007 ¶¶21-25; *see also* Ex-1008 ¶¶13, 38-39, 53 (disclosing a power management unit of a system that allocates a fixed power budget to itself).

129. Thus, it is my opinion that Nussbaum teaches that the third domain to operate at a fixed power budget.

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

130. In my opinion, Nussbaum teaches Element 13[A.ii(3)].

131. As shown in annotated Figure 1, Nussbaum explains that "performance analysis control logic 111" and "power allocation controller 109" (yellow) jointly

serve as a system agent to "analyze[] performance sensitivity of the cores and other computational units" and "control[] allocation of the Thermal Design Point (TDP) ***power headroom*** to the on-die or on-platform components." Ex-1007 ¶20; *see also id*. ¶¶19, 21-25, 39. Nussbaum also discloses determining an "***available power headroom***" that can be dynamically allocated and explains that "[a]ny ***change in the state*** of the on-die components ***triggers an update*** of" this available power headroom, confirming that this power headroom is a variable. *Id*. ¶¶36, 39.

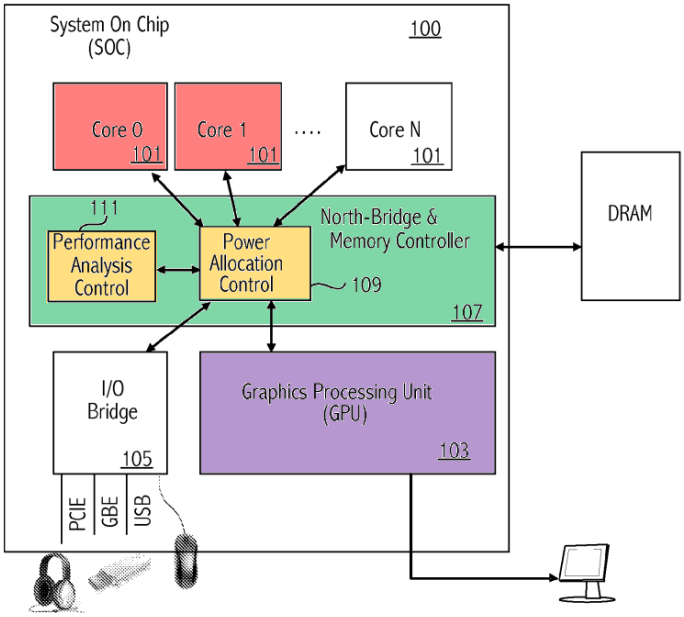


FIG. 1

Ex-1007 Fig. 1 (annotated).

132. Nussbaum further describes ***selectively distributing*** this power headroom ***among*** the computational units based on the computational units' sensitivities to frequency changes. Ex-1007 ¶19; *see also id*. ¶¶20-25. Nussbaum also provides an example where "boost sensitivit[ies]" of two computational units

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

133. Thus, it is my opinion that Nussbaum discloses a power sharing logic to dynamically allocate a variable power budget between the first and second domains.

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

134. In my opinion, Nussbaum teaches Element 13[A.ii(4)].

135. Nussbaum describes ***selectively distributing*** a power headroom ***among*** its computational units based on the computational units' sensitivities to frequency changes, and provides an example where "boost sensitivit[ies]" of two computational units are used to "determine how to ***allocate power between***" them. Ex-1007 ¶¶19, 25; *see also id*. ¶¶20-24. Nussbaum explains that the "boost

sensitivity" indicates a "performance sensitivity of each computational unit to frequency change, and/or other change in performance capability," and further explains that the "computational units," for which boost sensitivities are determined, may include the cores operating in the first domain as well as the GPU operating in the second domain. *Id*. Accordingly, Nussbaum discloses allocating power between the first and second domains based at least in part on a boost sensitivity of a core operating in the first domain (i.e., a first power sharing value for the first domain) and a boost sensitivity of the GPU operating in the second domain (i.e., a second power sharing value for the second domain).

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

values associated with boost sensitivities. *Id*. ¶¶26-27, 30-31. Nussbaum therefore teaches that the first power sharing value for the first domain is stored in a first storage (corresponding to a first register that stores the boost sensitivity value for a core operating in the first domain) and the second power sharing value for the second domain is stored in a second storage (corresponding to a second register that stores the boost sensitivity value for the GPU operating in the second domain). Ex-1007 ¶¶25-32, Fig. 4; *see also* Ex-1001 4:1-20 (referring to registers as an example of where power sharing values may be stored).

137. Thus, it is my opinion that Nussbaum teaches a power sharing logic to dynamically allocate a variable power budget between the first and second domains based at least in part on a first power sharing value for the first domain stored in a first storage and a second power sharing value for the second domain stored in a second storage.

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

138. In my opinion, Nussbaum teaches Element 13[B].

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

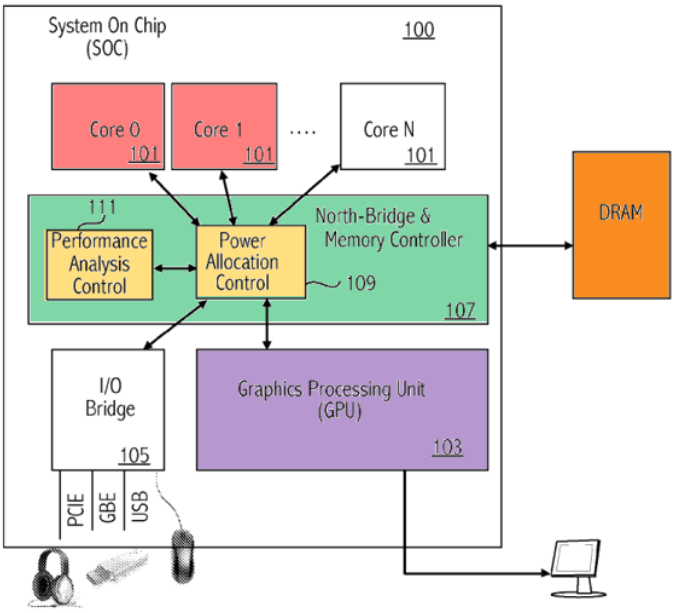


FIG. 1

Ex-1007 Fig. 1 (annotated).

140. Thus, it is my opinion that Nussbaum teaches a dynamic random access memory (DRAM) coupled to the multicore processor.

2. Dependent claims 14-15 and 17-18

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

141. In my opinion, Nussbaum teaches Claim 14.

142. Nussbaum discloses determining an "***available power headroom***" that can be dynamically allocated and explains that "[a]ny ***change in the state*** of the on-die components ***triggers an update*** of" this available power headroom. Ex-1007

¶¶36, 39. Nussbaum further explains that "change of state that triggers the update" may include, e.g., "a change in performance or power state or change in application/workload activity," "a process context change," or "a performance state change," confirming that the "available power headroom" is updated at run time for the current time interval. *Id*.

143. And as described above in Sections VIII.C.1.e-f, Nussbaum also discloses allocating a first portion of the variable power budget (the available power headroom) to the first domain according to the first power sharing value, and allocating a second portion of the variable power budget to the second domain according to the second power sharing value. *See also* Ex-1007 ¶¶23, 30, 33.

144. Thus, it is my opinion that Nussbaum teaches that the power sharing logic to determine the variable power budget for a current time interval and to allocate a first portion of the variable power budget to the first domain according to the first power sharing value, and to allocate a second portion of the variable power budget to the second domain according to the second power sharing value.

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

145. In my opinion, Nussbaum teaches Claim 15.

146. As shown in Figure 6, Nussbaum discloses providing "***the power headroom*** made available through throttling" to "bounded computational unit(s)." Ex-1007 ¶47. Because Nussbaum does not subdivide this "power headroom," a POSITA would have understood that Nussbaum contemplates providing the entire "power headroom made available through throttling" to "bounded computational unit(s)." *See*, *e.g.*, Ex-1005 ¶34 (explaining that "a single power plane may obtain the entire budget"); Section VIII.A.2.e (explaining that a POSITA would have understood to allocate an entire budget to one of the domains performing intensive workload). Nussbaum also recognizes that "[m]ost real-world applications are either CPU or GPU-bounded." *Id*. ¶¶22, 44, 46. Accordingly, Nussbaum explicitly contemplates providing the entire "***power headroom*** made available through throttling" to CPU-bounded computational unit(s) when applications are CPU-bounded, including cores operating in the first domain. And when applications become GPU-bounded, the entire power headroom is made available to GPU-bounded computational unit(s), including GPU operating in the second domain.

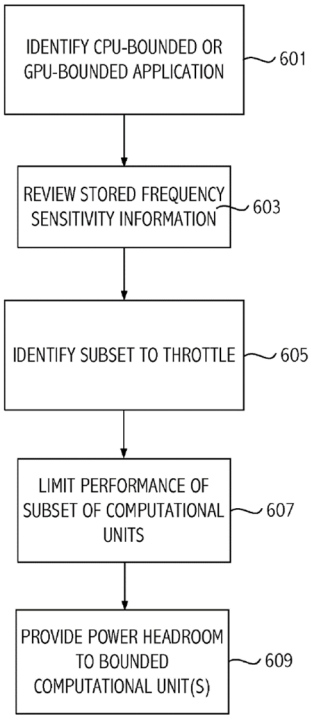


FIG. 6

Ex-1007 Fig. 6.

147. Thus, it is my opinion that Nussbaum teaches that the power sharing logic is to dynamically allocate substantially all of the variable power budget to the first domain for a first workload, and to dynamically allocate substantially all of the variable power budget to the second domain for a second workload executed after the first workload, notwithstanding that the lower-bound of "substantially all" may not be reasonably ascertainable in other contexts.

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

148. In my opinion, Nussbaum teaches Claim 17.

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

TABLE 3

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

150. Similarly, as shown in Table 4 below, Nussbaum discloses "four power states" for the GPU operating in the second domain, and a POSITA would have understood that the operational point corresponding to "GPU_P3" represents a second minimum reservation value for the second domain because it corresponds to the lower performance state the GPU in the second domain may be utilized to perform its operations. Ex-1007 ¶¶34-38.

TABLE 4

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

151. Thus, it is my opinion that Nussbaum teaches that the power sharing logic is to further allocate the first portion of the variable power budget according to a first minimum reservation value for the first domain and allocate the second portion of the variable power budget according to a second minimum reservation value for the second domain.

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

152. In my opinion, Nussbaum teaches Claim 18.

153. As shown in Figure 6 below, Nussbaum discloses allocating its variable power budget based on an output value of step 601 that identifies whether an application is "CPU-bounded" or "GPU-bounded." Ex-1007 ¶¶44-48. Nussbaum explains that if the output value of step 601 identifies an application as "GPU-bounded," then "CPU cores may be throttled" to "release power margin available to the GPU." *Id*. ¶44. Thus, a POSITA would have understood that an output value of step 601 being an identification of a "GPU-bounded" application is a preference value to favor the second domain (which includes the GPU) over the first domain (which includes the cores).

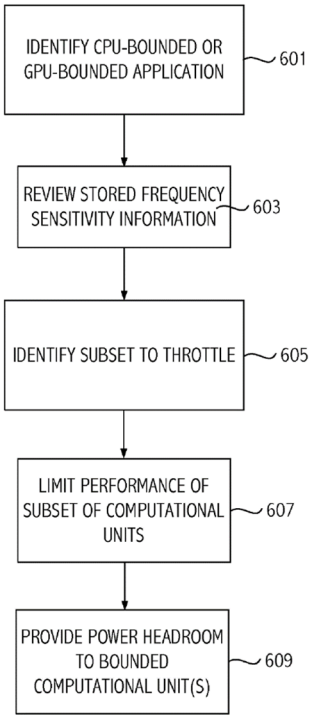


FIG. 6

Ex-1007 Fig. 6.

154. Thus, it is my opinion that Nussbaum teaches that the power sharing logic is to further allocate the variable power budget according to a preference value, the preference value to favor the second domain over the first domain.

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

1. Dependent claim 16

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

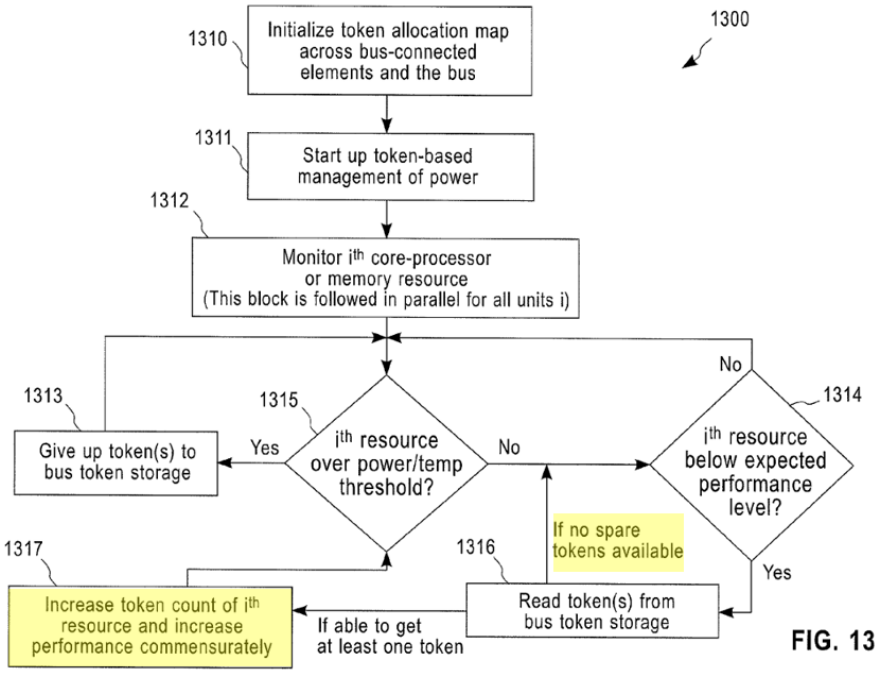
155. In my opinion, Nussbaum in view of Bose teaches Claim 16.

156. Nussbaum teaches incrementing the first power sharing value when a request for a higher frequency for the first domain is not granted, and incrementing the second power sharing value when a request for a higher frequency for the second domain is not granted. For example, Nussbaum "determines ***boost sensitivity*** of a computational unit in response to a process context change associated with the computational unit." Ex-1007 ¶7. Nussbaum explains that the boost sensitivity of a core "represents the sensitivity of the core, running that particular process context, to a change in frequency." *Id*. ¶27; *see also id*. ¶32. Nussbaum further discloses that "[t]he boost sensitivity for each core is tied to the current processor context, [and] when the context changes, the ***sensitivity is re-evaluated***." *Id*. ¶30. Nussbaum also explains that boost sensitivity may "expire[] for each context based on a fixed or programmable timer," and in some implementations, "both a timer and context switch, whichever occurs first, are used to initiate the boost sensitivity reevaluation."

Id. Thus, a POSITA would have understood that Nussbaum explicitly contemplates changing (e.g., incrementing) boost sensitivities of computational units, including computational units whose requests for a higher frequency were not granted. A POSITA would have also understood that boost sensitivity reevaluation is applicable equally to the first and second domains, rendering it obvious for Nussbaum to increment the first power sharing value when a request for a higher frequency for the first domain is not granted, and to increment the second power sharing value when a request for a higher frequency for the second domain is not granted.

157. Moreover, a POSITA seeking to implement the teachings of Nussbaum would also have found this feature obvious in view of Bose. Like Nussbaum, Bose discloses "power management of a system of connected components … wherein each component is assigned a power budget as determined by a number of allocated tokens in the token allocation map." Ex-1008, Abstract. Bose utilizes a "token management unit (TMU)" to manage allocation of tokens to "token-managed units," such as "processor resources," and explains that "[e]ach managed component, in response to its own operating conditions (as evidenced by its utilization markers) can either attempt to donate one or more token(s)" or "read (receive) one or more token(s)," e.g., by "request[ing] for token read/write operations … to be serviced." *Id*. ¶¶38-39, 68. Bose further discloses that if the performance level of a particular resource is "below a predefined minimum performance," a process is initiated to "***try***

and read additional tokens from [a] centralized bus token storage … to boost up performance" of that particular resource, e.g., "by ***increasing its voltage and frequency***, if allowed by the system architecture." *Id*. ¶86. And as shown in annotated Figure 13 below, Bose explains that if no additional tokens can be allocated to the particular resource at the moment (i.e., a request for a higher frequency is not granted), that particular resource can wait for its token count to be increased in the next iteration of the process. *Id*. ¶86, Fig. 13.



Ex-1008 Fig. 13 (annotated).

158. A POSITA would have been motivated to implement Nussbaum's power allocation operations with Bose' teaching to increment a boost sensitivity of a particular domain (e.g., the first or the second domain) when a request for a higher

frequency for that particular domain is not granted, especially if that particular domain is operating below an expected performance level, so as to avoid continued denial of that particular domain's request for a higher frequency and to avoid starvation of that domain. *See* Ex-1008 ¶86, Fig. 13; *see also* Ex-1025, 2:2-3:2, 5:66-67, 6:25-31, 6:32-47 (describing re-scheduling of resource allocations "when a request cannot be fulfilled" such that "the requesting application has a higher priority than at least some of the other executing applications" as a "sensible way" to avoid starvation). A POSITA would have had a reasonable expectation of success because Nussbaum explicitly contemplates re-evaluating its boost sensitivities. Ex-1007 ¶30. A POSITA would also have had a reasonable expectation of success because such operations have already been implemented in Bose.

159. Thus, it is my opinion that Nussbaum in view of Bose teaches that the power sharing logic is to increment the first power sharing value when a request for a higher frequency for the first domain is not granted, and to increment the second power sharing value when a request for a higher frequency for the second domain is not granted.

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

1. Independent claim 13

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

160. In my opinion, Conroy teaches the preamble of Claim 13, to the extent it is limiting.

161. For example, Conroy discloses "methods and apparatuses to dynamically redistribute power in a ***system*** that includes a plurality of subsystems." Ex-1006, Abstract.

162. Thus, it is my opinion that Conroy discloses a system as described in the preamble.

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

163. In my opinion, Conroy teaches Element 13[A.i].

164. Conroy teaches a multicore processor. As shown in annotated Figure 37 below, Conroy discloses "system 3700," which a POSITA would have understood to be a processor. Moreover, to the extent it is argued that a processor must be implemented on a chip, a POSITA seeking to implement the teachings of Conroy would have found that to be obvious because doing so would improve the reliability and power efficiency of system 3700 while also reducing its size and weight. *See*, *e.g.*, Ex-1007, Fig. 1 (implementing a system similar to system 3700

as a System on Chip (SOC)); Ex-1005, Fig. 1. Indeed, the '833 Patent itself recognizes that "[o]ver time, processor designs have evolved from a collection of independent integrated circuits … to multicore processors that include multiple processor cores within a single IC package." Ex 1001, 1:13-19. The '833 Patent also defines "processor" broadly to include implementations where function units are "formed on a single semiconductor die" and "other implementations" that "can be realized by a multi-chip package in which different domains can be present on different semiconductor die of a single package." Ex-1001, 1:56:-2:4.

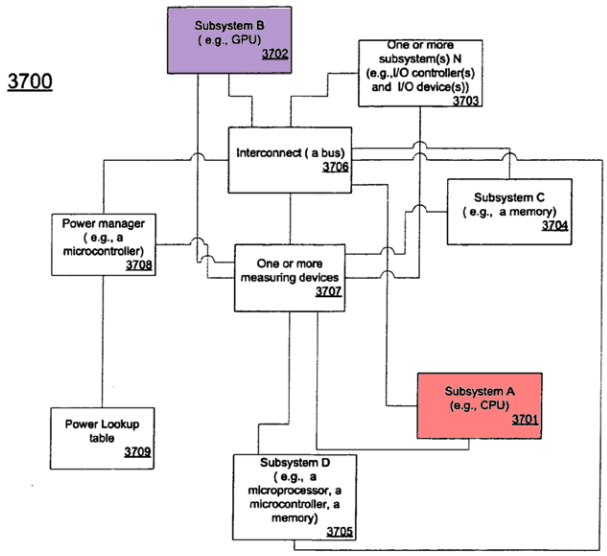


FIG. 37

Ex-1006 Fig. 37 (annotated).

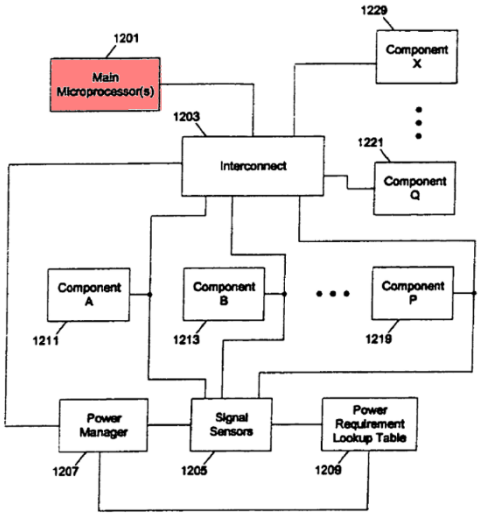


Fig. 12

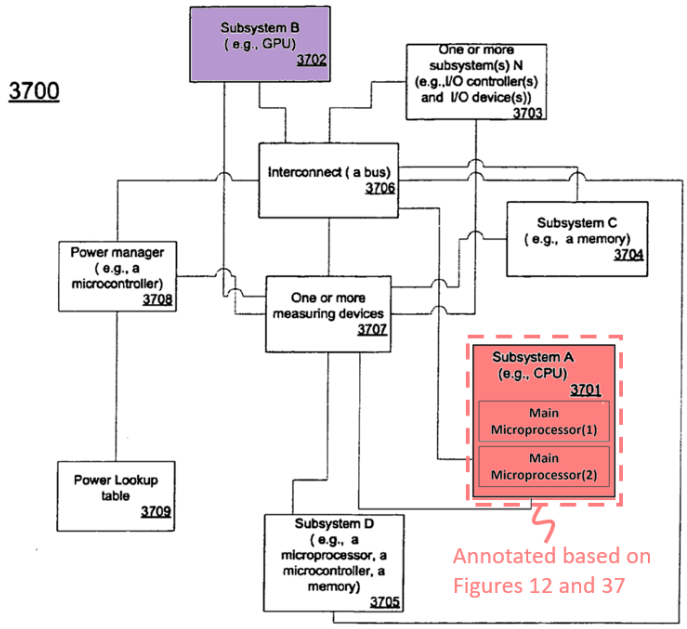
Ex-1006 Fig. 12 (annotated).

165. Conroy also teaches that the multicore processor has a first domain including a plurality of cores. Conroy states that system 3700 includes "subsystem A (e.g., CPU)" 3701 (red), and that "[c]omponents of the system 3700, including

processors [and] ***microcontrollers***,… are described in detail … with respect to FIGS. 1-17." Ex-1006 ¶225. One of these figures, namely Figure 12 (annotated above), depicts a CPU (red) as a plurality of "main microprocessor***(s)***" 1201. *Id*. ¶167.

166. Thus, a POSITA would have understood that Conroy explicitly contemplates its Subsystem A 3701 (red) to include a plurality of "main microprocessor***(s)***," which correspond to a plurality of cores, as shown in the composite Figure 37′ below. Ex-1006 ¶¶50, 65-66, 167, 188-194, 225, Figs. 12, 37. Furthermore, Conroy discloses using "power manager" to determine "***a throttle setting***" for the plurality of "main ***microprocessor(s)***," confirming that Conroy explicitly contemplates having a plurality of cores operating in the same domain with the same "throttle setting." *Id*. ¶¶167, 225. A POSITA would have understood that such a "throttle setting" is used to control the power settings of the plurality of cores, including the operating voltage and/or frequency of the plurality of cores. *See*, *e.g.*, *id*. ¶50 ("For example, the CPU processor of the computer system may be set to run under different core voltages and/or different core frequencies such that the system can be fully functional but at different performance levels to trade power usage and computing performance."); *see also id*. ¶¶65-66; Ex-1005 ¶¶24, 28 (describing "voltage and/or frequency changes" as examples of changing "power setting(s)"); Ex-1007 ¶¶4, 37 ("The performance levels are typically determined by

voltage/frequency combinations used by the processor."). Because Conroy explicitly contemplates having the plurality of cores operating at the same "throttle setting," a POSITA would have understood that to mean that the plurality of cores are operating at the same voltage and frequency point. Ex-1006 ¶¶167, 225; *see also id*. ¶¶50, 65-66; Ex-1005 ¶¶24, 28; Ex-1007 ¶¶4, 37.



Composite Figure 37'

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

would have understood that Conroy's GPU (purple) is set to operate at a power setting determined based on a load profile, and, as discussed above, a POSITA would have understood such a power setting to include an operating voltage and/or frequency for the GPU. *Id*.; *see also id*. ¶¶50, 65-66; Ex-1005 ¶¶24, 28; Ex-1007 ¶¶4, 37. Thus, a POSITA would have understood that Conroy discloses a GPU that operates in a second domain.

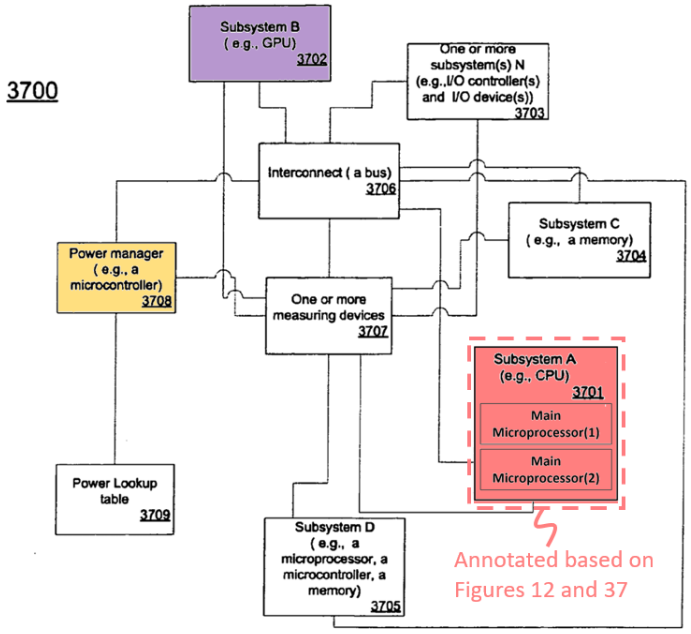
168. Thus, it is my opinion that Conroy discloses a multicore processor having a first domain including a plurality of cores, and a second domain including a graphics engine.

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

169. In my opinion, Conroy teaches Element 13[A.ii(1)].

170. Conroy discloses that system 3700 includes "power manager 3708, e.g., a microcontroller." Ex-1006 ¶225. Conroy explains that its "power manager[] (e.g., implemented as a microcontroller) is used to dynamically determine the throttle settings of the system to balance the performance requirement and the power usage limit," and further explains that its microcontroller "used to budget the power usage dynamically … can be considered as part of the non-throttled components." *Id*. ¶¶94, 168. Thus, Conroy teaches a third (non-throttled) domain including a power manager implemented as a microcontroller (yellow). *Id*. ¶¶94, 168, 225, Figs. 4, 12,

37. Moreover, the power manager (yellow) is a "system agent" similar to that described in the '833 Patent, which explains that a "system agent" is a component that controls power supply to processor components. *See*, *e.g.*, Ex-1001, 9:16-21 ("[S]ystem agent domain 450 may execute at a fixed frequency and may remain powered on at all times to handle power control events and power management such that domains 410 and 420 can be controlled to dynamically enter into and exit low power states."), 9:48-50 ("[S]ystem agent domain 450 may include a power control unit 455 which can include a power sharing logic 459 in accordance with an embodiment of the present invention.").



Composite Figure 37'

171. Thus, it is my opinion that Conroy discloses a multicore processor having a third domain including system agent circuitry (power manager implemented as a microcontroller).

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

172. In my opinion, Conroy teaches Element 13[A.ii(2)].

173. Conroy states that microcontroller "used to budget the power usage dynamically … can be considered as part of the ***non-throttled*** components," which are "not throttled to trade performance for power usage." Ex-1006 ¶¶81, 94. Conroy also discloses determining a power budget for the non-throttled components by "adding together the maximum powers which could be consumed by the non-throttled subsystems (components)," and explains that this calculation "can be done when the system is designed; and the result can be a ***constant***." *Id*. ¶150. Conroy therefore teaches allocating a fixed power budget (a ***constant*** that is sufficient to accommodate the sum of "the maximum powers which could be consumed by the non-throttled" components) to the non-throttled components.

174. Thus, it is my opinion that Conroy teaches that the third domain to operate at a fixed power budget.

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

175. In my opinion, Conroy teaches Element 13[A.ii(3)].

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

177. Thus, it is my opinion that Conroy discloses a power sharing logic to dynamically allocate a variable power budget between the first and second domains.

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

178. In my opinion, Conroy teaches Element 13[A.ii(4)].

179. Conroy discloses that the power sharing logic allocates power between the first and second domains based at least in part on first and second power sharing values for the first and second domains. For example, Conroy discloses implementations in which "a workload of a subsystem determines the amount of power used by the subsystem" and explains that the power may be "redistributed in an asymmetric fashion, tracking the workloads of each of the subsystems." Ex-1006 ¶¶196-197. During a "CPU-heavy load profile," when the workload in the first domain is intensive, power may be distributed such that "CPU consumes about 75%,

while GPU consumes about 25% of the power allocated for the system," and vice versa. *Id*. ¶203. Conroy further discloses selecting a "***power weighting arrangement among the subsystems*** … based on the load profile" and states that "the power weighting arrangement … may be a power distribution table ***selected out of a plurality of power distribution tables***." *Id*. ¶¶6, 200. Examples of power distribution tables are illustrated in Figures 21A-C and 22, copied below, confirming that Conroy discloses dynamically allocating a power budget according to first and second sharing policy values (power weighting arrangement settings selected from a power distribution table based on load profile) for the first and second domains. *Id*. ¶¶6, 200, Figs. 21A-C, 22.

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

2100

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

2105

Fig. 21 A

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

2110

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

2105

Fig. 21 B

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

2120

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

2105

Fig. 21 C

Load Profile Kn

2200

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

2208

Fig. 22

Ex-1006 Figs. 21A-21C, 22.

180. Conroy also teaches that first and second power sharing values are stored in first and second storage. For example, Conroy explains that its "power distribution table" is described "with respect to FIGS. 7, 21, and 22." Ex-1006 ¶225. Conroy's Figure 7, copied below, shows a power distribution table that stores power settings for various system components, and Conroy describes an example where these power settings are stored in registers. *Id.* ¶97 ("For example, the microcontroller may store [a] determined throttle setting in a register.").

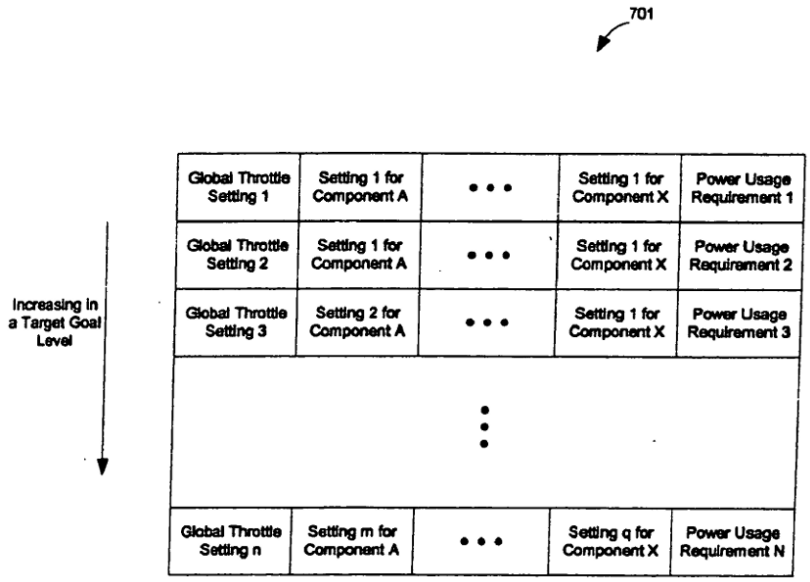


Fig. 7

Ex-1006 Fig. 7.

181. Similarly, Figures 21 and 22, above, also depict power distribution tables that store power settings for various system components. Accordingly, a POSITA seeking to implement the teachings of Conroy would have found it obvious to use registers to store the various power settings depicted in Conroy's Figures 7, 21, and 22. *Id.* ¶¶97, 225.

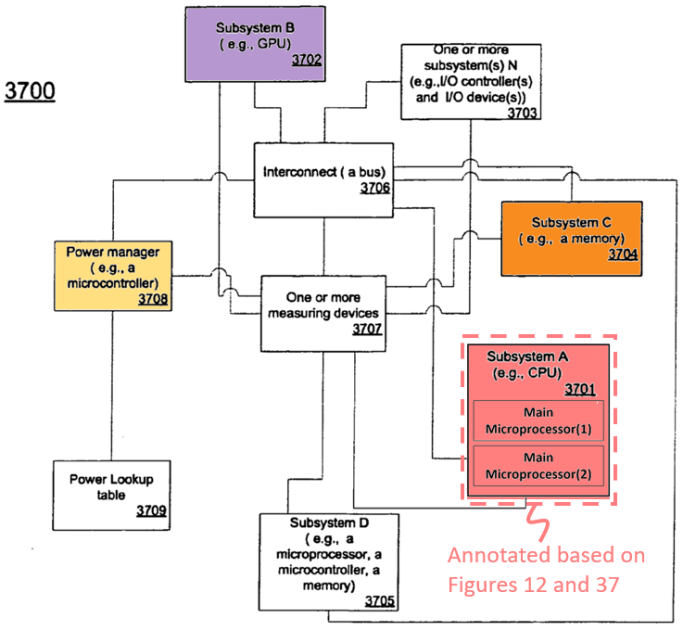
182. Thus, it is my opinion that Conroy teaches a power sharing logic to dynamically allocate a variable power budget between the first and second domains based at least in part on a first power sharing value for the first domain stored in a first storage (corresponding to a first register that stores a power setting for a CPU

operating in a first domain) and a second power sharing value for the second domain stored in a second storage (corresponding to a second register that stores a power setting for a GPU operating in a second domain).

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

183. In my opinion, Conroy teaches Element 13[B].

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



Composite Figure 37′

185. Thus, it is my opinion that Conroy teaches a dynamic random access memory (DRAM) coupled to the multicore processor.

2. Dependent claims 14-15 and 17-18

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

186. In my opinion, Conroy teaches Claim 14.

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

188. And as discussed in Sections VIII.E.1.e-f, Conroy also discloses allocating a first portion of the variable power budget (the dynamically determined "allowable power budget") to the first domain according to the first power sharing value, and allocating a second portion of the variable power budget to the second domain according to the second power sharing value. *See also* Ex-1006 ¶¶8-9, 122, 151, 154, 177, 202-203.

189. Thus, it is my opinion that Conroy teaches that the power sharing logic to determine the variable power budget for a current time interval and to allocate a first portion of the variable power budget to the first domain according to the first power sharing value, and to allocate a second portion of the variable power budget to the second domain according to the second power sharing value.

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

190. In my opinion, Conroy teaches Claim 15.

191. Conroy discloses implementations in which "a workload of a subsystem determines the amount of power used by the subsystem" and recognizes that "program development tools and scientific applications present a ***high load to the CPU***, but ***almost no load to the GPU***." Ex-1006 ¶196. Conroy also recognizes that many applications "present an alternating high workload to the CPU and to the GPU that results in an ***alternating asymmetric load profile*** of the system." *Id*. Conroy explains that for a "CPU-heavy load profile," CPU may "consume[] a ***substantially bigger*** share of the total system's power, while a graphics processor is ***hardly used at all***," and that for a "GPU-heavy load profile," GPU may "consume[] a ***substantially bigger*** share of the total system's power, while a CPU is ***hardly used***

at all." *Id*. ¶203. Thus, Conroy explicitly contemplates redistributing substantially all of its variable power budget to the CPU when the GPU has "almost no load" or is "hardly used at all," and vice versa.

192. Thus, it is my opinion that Conroy teaches that the power sharing logic is to dynamically allocate substantially all of the variable power budget to the first domain for a first workload, and to dynamically allocate substantially all of the variable power budget to the second domain for a second workload executed after the first workload, notwithstanding that the lower-bound of "substantially all" may not be reasonably ascertainable in other contexts.

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

193. In my opinion, Conroy teaches Claim 17.

194. Conroy discloses a "power usage range" between a "minimum possible power consumption" and a "maximum power consumption." Ex-1006 ¶53, Fig. 1. Conroy also explains that even when a subsystem has the lowest power allocation utilization of 0, Conroy's power manager still allocates power to that subsystem to maintain the subsystem in a "low power mode." *Id*. ¶198. A POSITA would have understood that the reason to have a low power mode is so that a component can be

quickly brought to an active state, rather than started from a cold, completely-off state, and that in such a low power mode there would also be leakage resulting in power consumption. Conroy further discloses sensing, predicting, or estimating "the power used by ***each of*** the subsystems," confirming that the "minimum possible power consumption" represents an allocation that, at a minimum, allocates portions of the variable power budget to satisfy the minimum reservation values for ***each of*** the subsystems operating in the first and second domains. *Id*. ¶¶6, 8, 198, 202-206, Abstract.

195. Thus, it is my opinion that Conroy teaches that the power sharing logic is to further allocate the first portion of the variable power budget according to a first minimum reservation value for the first domain and allocate the second portion of the variable power budget according to a second minimum reservation value for the second domain.

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

196. In my opinion, Conroy teaches Claim 18.

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

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

198. Thus, it is my opinion that Conroy teaches that the power sharing logic is to further allocate the variable power budget according to a preference value, the preference value to favor the second domain over the first domain.

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

1. Dependent claim 16

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

199. In my opinion, Conroy in view of Bose teaches Claim 16.

200. Conroy discloses incrementing the first and second power sharing values for the first and second domains. As shown in Figure 22, Conroy recognizes that system performance "may move up and down across column 2206," and "if the performance of the system ***requires an increase*** in total power, the performance

moves up across the column e.g., from system's setting M to system's setting 1." Ex-1006 ¶204.

Load Profile Kn

2200

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

2208

Ex-1006, Fig. 22.

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

202. Moreover, a POSITA seeking to implement the teachings of Conroy would also have found this feature obvious in view of Bose. Like Conroy, Bose discloses "power management of a system of connected components … wherein each component is assigned a power budget as determined by a number of allocated tokens in the token allocation map." Ex-1008, Abstract. And as discussed in Section VIII.D.1.a, Bose discloses instructing a particular component to wait for its token count to be increased in the next iteration of the process if its request for a higher frequency is not granted. A POSITA would have been motivated to implement

Conroy's power manager with Bose's teaching to not grant a subsystem's request to increase the system's setting immediately and, instead, wait till the next iteration to do so in order to keep the power supplied to the various subsystems stable within each integration. A POSITA would have had a reasonable expectation of success in doing so given that Conroy itself explicitly contemplates such implementations. Ex-1006 ¶130. A POSITA would also have had a reasonable expectation of success because such operations have already been implemented in Bose.

203. Thus, it is my opinion that Conroy in view of Bose teaches that the power sharing logic is to increment the first power sharing value when a request for a higher frequency for the first domain is not granted, and to increment the second power sharing value when a request for a higher frequency for the second domain is not granted.

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

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

204. In addition to the reasons set out below with respect to specific claim elements, a POSITA would have been motivated to combine the teachings of Conroy and Finkelstein, and would have had a reasonable expectation of success in doing so, because each relates to the same well-known issues. Specifically, both relate to

and describe similar methods for managing power allocation in a system with multiple cores or subsystems.

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

206. Like Conroy, Finkelstein discloses distributing a power budget and provides several formulae by which a power management logic may dynamically allocate the power budget between multiple power planes. Ex-1005 ¶¶28-36. And, again like Conroy, Finkelstein also discloses sharing an "energy budget" so that "unused processor core power" can be used for other cores that may be performing more "intensive workload." *Id*. ¶26.

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

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

2. Independent claim 1

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

209. In my opinion, Conroy alone or in view of Finkelstein teaches the preamble of Claim 1, to the extent it is limiting.

210. As shown in Figure 37, Conroy discloses "system 3700" that includes a CPU, a GPU, and a power manager. Ex-1006 ¶225. A POSITA would have

understood such a system to be a processor. *See* Section VIII.E.1.b. And to the extent it is argued that a processor must be implemented on a chip, a POSITA seeking to implement the teachings of Conroy would have found that to be obvious. *Id*.

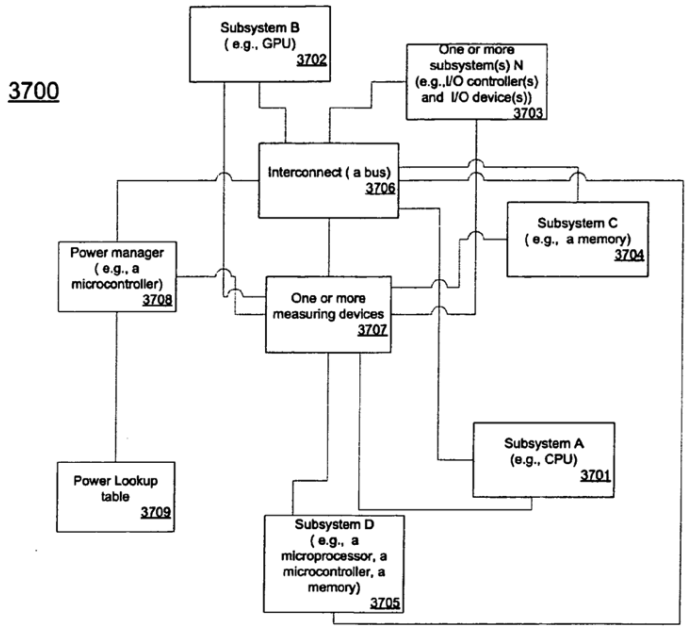


FIG. 37

Ex-1006 Fig. 37.

211. Moreover, a POSITA would also have found it obvious to implement system 3700 on a chip in view of Finkelstein. Like Conroy, Finkelstein discloses a system that includes a CPU, a graphics circuit, and a power manager. Ex-1005 ¶¶11, 42-43, 48-50, Figs. 1, 6, 7. Finkelstein states that its components "may be implemented on a single integrated circuit (IC) chip" and refers to such a single IC chip as a "processor." *Id*. A POSITA seeking to implement the teachings of Conroy

would have been motivated to implement Conroy's system 3700 on a single IC chip to improve its reliability and power efficiency while also reducing its size and weight. Additionally, a POSITA would have had a reasonable expectation of success in doing so because such implementations have already been provided in Finkelstein.

212. Thus, it is my opinion that Conroy alone or in view of Finkelstein teaches a processor as described in the preamble.

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

213. In my opinion, Conroy teaches Element 1[A].

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

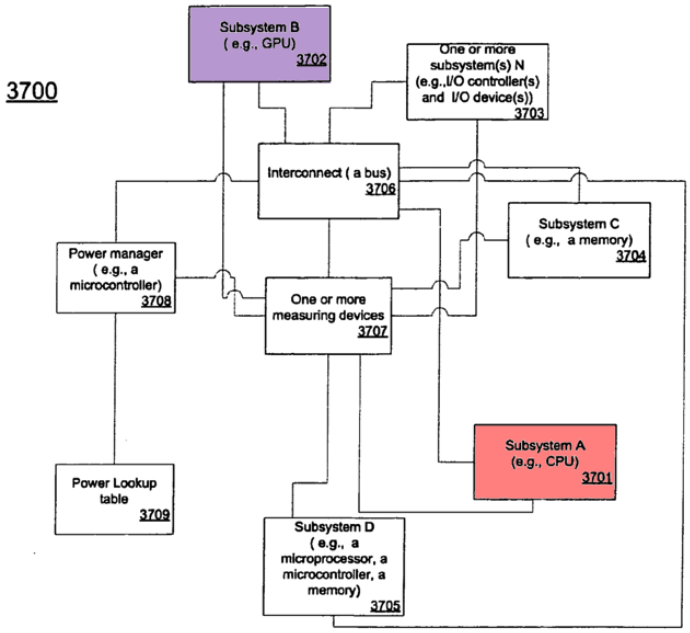


FIG. 37

Ex-1006 Fig. 37 (annotated).

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

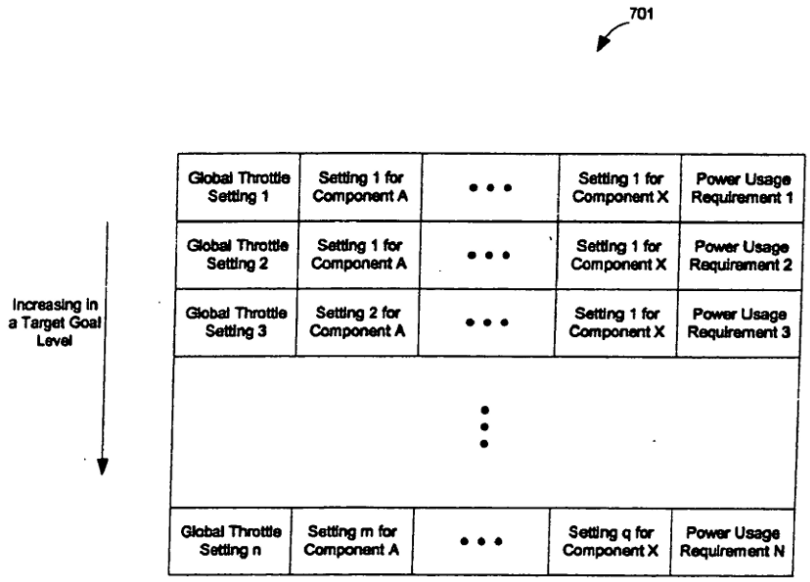


Fig. 7

Ex-1006 Fig. 7.

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

domain) and the GPU (in a second domain) at independent voltages and frequencies so that they can operate at independent settings for an increased ability to balance performance and control. *Id.*; *see also id.* ¶120 ("[M]ultiple components have ***independent*** throttle settings."); Ex-1027 at 35-40 (describing "independent" voltage and frequency control for each core or domain).

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

2100

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

2105

Fig. 21 A

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

2110

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

2105

Fig. 21 B

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

2120

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

2105

Fig. 21 C

Load Profile Kn

2200

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

2208

Fig. 22

Ex-1006 Figs. 21A-21C, 22 (annotated).

217. Moreover, a POSITA would also have understood that different power settings correspond to different operating voltages and/or frequencies. Ex-1006 ¶50 ("For example, the CPU processor of the computer system may be set to run under different core voltages and/or different core frequencies such that the system can be fully functional but at different performance levels to trade power usage and computing performance."); *see also id*. ¶¶65-66; Ex-1005 ¶¶24, 28 (describing "voltage and/or frequency changes" as examples of changing "power setting(s)"); Ex-1007 ¶¶4, 37 ("The performance levels are typically determined by voltage/frequency combinations used by the processor.").

218. Thus, it is my opinion that Conroy teaches a first domain and a second domain, each of the first and second domains to operate at an independent voltage and frequency.

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

219. In my opinion, Conroy alone or in view of Finkelstein teaches Element 1[B].

220. As shown in annotated Figure 37 below, Conroy discloses "system 3700" that includes "interconnect (e.g., a bus)" 3706 (green) coupled to a CPU (red) operating in a first domain and a GPU (purple) operating in a second domain. Conroy explains that "[c]omponents of the system 3700, including … ***buses***, … are

described in detail … with respect to FIGS. 1-17." Ex-1006 ¶225. For example, interconnect bus 1702, depicted in Figure 17, may "include one or more buses connected to each other through various bridges, ***controllers*** and/or adapters as is well known in the art." *Id*. ¶187. A POSITA would have recognized that memory controllers are "controllers … well known in the art," and, because Conroy explicitly depicts system 3700 that includes "subsystem C (e.g., ***memory***)" 3704 (orange), a POSITA seeking to implement the teachings of Conroy would have found it obvious to implement interconnect 3706 (green) by including a bus connected to a memory controller, which is in turn coupled to the CPU (red) operating in the first domain and the GPU (purple) operating in the second domain. *Id*. ¶¶187, 225, Figs. 17, 37. A POSITA would have understood that a memory controller may also be integrated into one or more subsystems of system 3700 (e.g., Subsystems A, B, or C) and thus not shown explicitly in the figure.

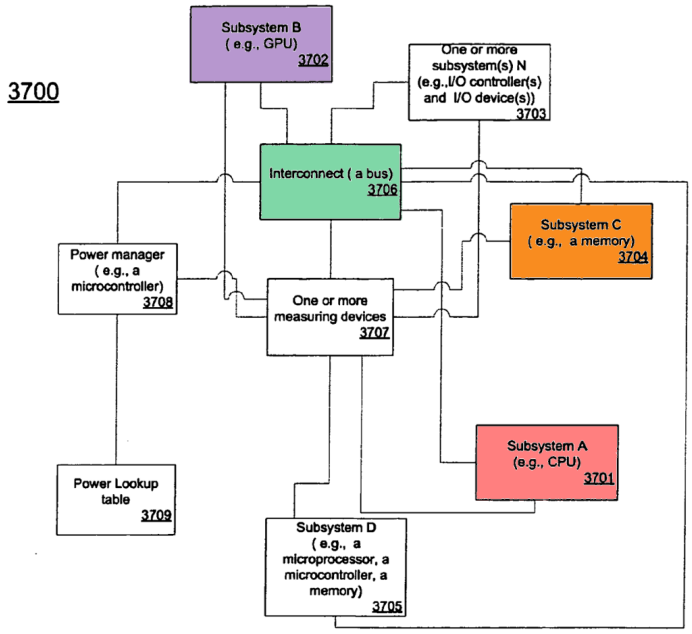


FIG. 37

Ex-1006 Fig. 37 (annotated).

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

222. Thus, it is my opinion that Conroy alone or in view of Finkelstein teaches a memory controller coupled to the first and second domains.

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

223. In my opinion, Conroy teaches Element 1[C].

224. As shown in annotated Figure 37, Conroy discloses "interconnect (e.g., a bus)" 3706 (green) that "interconnects … various components together." Ex-1006 ¶¶187, 225. A POSITA would have understood Conroy as teaching an "interface" because an "interconnection" that interconnects the various components of system 3700 together serves as an interface between these components. *Id*. ¶¶187, 225. Indeed, the '833 Patent explains that an "interface" enables "interconnection" between the processor cores and other circuitry. *See, e.g.,* Ex-1001, 9:58-60 ("Multiple interfaces … may be present to enable interconnection between the processor and other circuitry.").

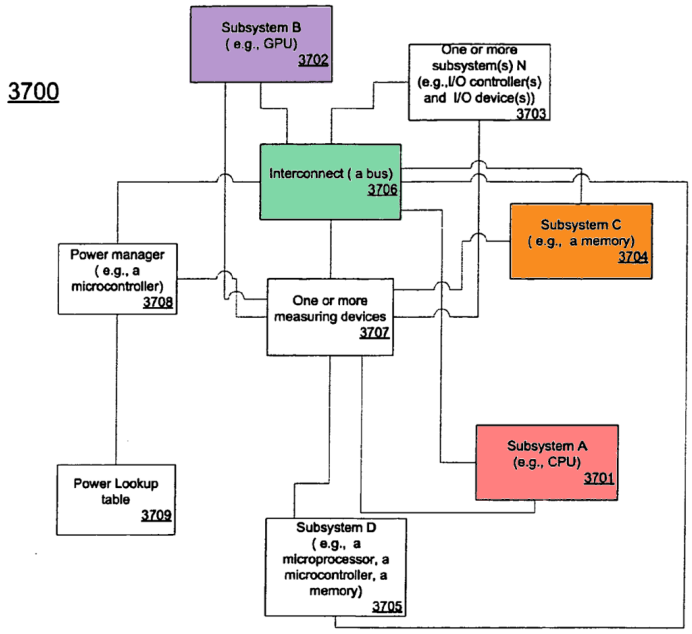


FIG. 37

Ex-1006 Fig. 37 (annotated).

225. Thus, it is my opinion that Conroy teaches at least one interface.

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

226. In my opinion, Conroy teaches Element 1[D.i].

227. As discussed in Section VIII.E.1.e, Conroy discloses a first logic to dynamically allocate a power budget for the processor between the first and second domains.

228. Conroy additionally discloses that this dynamic allocation of the power budget is performed "at run time." Conroy discloses implementations in which "a workload of a subsystem determines the amount of power used by the subsystem"

and explains that the power may be "redistributed in an asymmetric fashion, tracking the workloads of each of the subsystems." Ex-1006 ¶¶196-197. For example, during a "CPU-heavy load profile," when the workload in the first domain is intensive, power may be distributed such that "CPU consumes about 75%, while GPU consumes about 25% of the power allocated for the system." *Id*. ¶203. And, if the GPU operating in the second domain is engaged in intensive workload after the CPU, "the power of the system is shifted towards GPU" such that, for example, "GPU consumes about 75%, while CPU consumes about 25% of the power allocated for the system." *Id*. Because Conroy discloses dynamically allocating its power budget based on the workloads of each of the subsystems, a POSITA would have understood that such an allocation is performed "at run time" when the subsystems are processing their respective workloads.

229. Thus, it is my opinion that Conroy teaches a first logic to dynamically allocate a power budget for the processor between the first and second domains at run time.

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

230. In my opinion, Conroy in view of Finkelstein teaches Element 1[D.ii].

231. As discussed in Section VIII.E.1.f, Conroy discloses dynamically allocating a power budget according to first and second sharing policy values for the first and second domains.

232. A POSITA seeking to implement the teachings of Conroy would have found it obvious to make the first and second sharing policy values "controllable by user-level software" in view of Finkelstein. Like Conroy, Finkelstein discloses distributing a power budget between first and second domains. *See* Sections VIII.A.1.e-f. Finkelstein also discloses "user (or application defined) preferences" that describe how to distribute the power budget between the first and second domains. *Id*. A POSITA would have been motivated to combine the teachings of Conroy with the teachings of Finkelstein to provide a user with abilities to arrange or adjust the settings recorded in Conroy's power distribution tables, which are also used to describe how to distribute a power budget between first and second domains to provide an additional level of control over the system. *Id*.; *see also* Ex-1026, 11:13-27 (describing a "user preferences database" containing "parameters and procedures for how to apportion power among various devices"); Ex-1028 ¶17 ("The preference might be determined … based on a selection by a user of the system (e.g., via a graphical interface)."). Additionally, a POSITA would have had a reasonable expectation of success in doing so because Conroy itself contemplates certain levels of designer/user control over the recorded settings. Ex-1006 ¶¶121, 225 (explaining

that in certain implementations, the settings can be arranged "manually" by "the designer of the system"). Also, a POSITA would have had a reasonable expectation of success because Finkelstein already teaches such implementations with user-level control of power settings. Specifically, a POSITA implementing Conroy with the user-level control teachings of Finkelstein would have provided a user with the ability to input and/or change the values in Conroy's "power distribution tables" of Figures 21 and 22. This would have been nothing more than providing a routine user interface to edit and/or revise values in a routine table. *See*, *e.g.*, Ex-1026, 11:13-27; Ex-1028 ¶17.

233. Thus, it is my opinion that Conroy in view of Finkelstein teaches dynamic allocation of power the occurs according to a first sharing policy value for the first domain and a second sharing policy value for the second domain, the first and second sharing policy values controllable by user-level software.

3. Dependent claims 2-5 and 7

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

234. In my opinion, Conroy teaches Claim 2, for the reasons described above in Sections VIII.G.2.e-f.

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

235. In my opinion, Conroy alone or in view of Finkelstein teaches Claim 3.

236. As discussed in Section VIII.E.2.c, Conroy teaches that the first logic is to dynamically allocate the power budget further according to a first minimum reservation value for the first domain and a second minimum reservation value for the second domain, the first and second minimum reservation values controllable by the user-level software.

237. Moreover, a POSITA seeking to implement the teachings of Conroy would also have found this feature obvious in view of Finkelstein, which discloses dynamically allocating a power budget according to first and second minimum reservation values for the first and second domains, and that the first and second minimum reservation values are controllable by the user-level software. *See* Section VIII.A.2.b. A POSITA would have been motivated to combine the teachings of Finkelstein with the teachings of Conroy to ensure that Conroy's "minimum possible power consumption" takes into account the minimum possible power consumption of each subsystem (including subsystems operating in the first and second domains). Ex-1006 ¶¶6, 8, 53, 198, 202-206, Abstract, Fig. 1. Indeed, Conroy itself discloses

sensing, predicting, or estimating "the power used by ***each of*** the subsystems," confirming that the "minimum possible power consumption" takes into account the minimum reservation values for each of the subsystems operating in the first and second domains. *Id*. A POSITA would also have had a reasonable expectation of success in doing so because Conroy explicitly discloses meeting a "minimum possible power consumption" required by a system that includes subsystems such as CPU (operating in the first domain) and GPU (operating in the second domain). *Id*.

238. Thus, it is my opinion that Conroy alone or in view of Finkelstein teaches that the first logic is to dynamically allocate the power budget further according to a first minimum reservation value for the first domain and a second minimum reservation value for the second domain, the first and second minimum reservation values controllable by the user-level software.

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

239. In my opinion, Conroy alone or in view of Finkelstein teaches Claim 4, for the reasons described above for Claim 3.

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

240. In my opinion, Conroy in view of Finkelstein teaches Claim 5.

241. Conroy discloses that the first logic determines a power budget for a current time interval based at least in part on a power consumed in the previous time interval. For example, Conroy uses "the ***past power usage*** (e.g., measurements … at times T−(n−1)Δ, T−(n−2)Δ, … , T)" to "***dynamically determine the allowable power budget***" for time interval T+Δ, and allocates this "dynamically determined" power budget to different components of the system. Ex-1006 ¶¶12, 70-72, 80-83, 90, 120. A POSITA would have understood time interval T+Δ to be a "current time interval" because Conroy refers to the time intervals preceding T+Δ as being in the "***past***." *Id*. ¶¶70-72, Fig. 2.

242. Moreover, a POSITA seeking to implement the teachings of Conroy would have found it obvious to also take into account a power budget carried forward from a previous time interval and a power budget decay value in determining the power budget in view of Finkelstein. Like Conroy, Finkelstein discloses determining a power budget for a current time interval. *See* Section VIII.A.2.d. Finkelstein also discloses an energy budget equation based at least in part on a power

budget carried forward from a previous time interval, a power consumed in the previous time interval, and a power budget decay value. *Id*. A POSITA would have understood that Conroy may utilize an energy budget equation like the one disclosed in Finkelstein. For example, Conroy explains that in certain implementations, its "power budget … is further limited according to ***other conditions***." Ex-1006 ¶74. Thus, a POSITA would have been motivated to combine the teachings of Conroy with the teachings of Finkelstein and use Finkelstein's equation, which takes into account other conditions, to determine its power budget. Additionally, a POSITA would have had a reasonable expectation of success in using Finkelstein's equation in Conroy's power budget determination because both Conroy and Finkelstein are directed to managing power among multiple domains, both understand that changes to contemporary allocation may be needed based on preceding allocations, and both disclose affecting the current power budget according to the previous cycle. Ex-1005 ¶¶28-29; Ex-1006 ¶¶12, 70-74, 80-83, 90, 120.

243. Thus, it is my opinion that Conroy in view of Finkelstein teaches that the first logic is to determine the power budget for a current time interval based at least in part on a power budget carried forward from a previous time interval, a power consumed in the previous time interval, and a power budget decay value.

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

244. In my opinion, Conroy teaches Claim 7, for the reasons described above in Section VIII.E.2.b.

4. Independent claim 13

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

245. In my opinion, Conroy teaches the preamble of Claim 13, to the extent it is limiting, for the reasons described above in Section VIII.E.1.a.

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

246. In my opinion, Conroy alone or in view of Finkelstein teaches Element 13[A.i], for the reasons described above in Sections VIII.E.1.b, VIII.G.2.a.

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

247. In my opinion, Conroy teaches Element 13[A.ii(1)], for the reasons described above in Section VIII.E.1.c.

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

248. In my opinion, Conroy teaches Element 13[A.ii(2)], for the reasons described above in Section VIII.E.1.d.

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

249. In my opinion, Conroy teaches Element 13[A.ii(3)], for the reasons described above in Section VIII.E.1.e.

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

250. In my opinion, Conroy teaches Element 13[A.ii(4)], for the reasons described above in Section VIII.E.1.f.

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

251. In my opinion, Conroy teaches Element 13[B], for the reasons described above in Section VIII.E.1.g.

5. Dependent claims 14-15 and 17-18

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

252. In my opinion, Conroy teaches Claim 14, for the reasons described above in Section VIII.E.2.a.

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

253. In my opinion, Conroy teaches Claim 15, for the reasons described in Section VIII.E.2.b.

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

254. In my opinion, Conroy alone or in view of Finkelstein teaches Claim 17, for the reasons described in Sections VIII.E.2.c and VIII.G.3.b.

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

255. In my opinion, Conroy teaches Claim 18, for the reasons described in Section VIII.E.2.d.

IX. CONCLUSION

256. For the reasons set forth above, I believe that Claims 1-5, 7, and 13-18 of the ’833 Patent is unpatentable in view of the prior art.

257. In signing this declaration, I recognize that it 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 also recognize that I may be subject to cross-examination in the case and that cross-examination will take place in the United States. If cross-examination is required of me, I will appear for cross-examination within the United States during the time allotted for cross-examination.

258. I reserve the right to supplement my opinions in the future to respond to any arguments that the Patent Owner raises and to take into account new information as it becomes available to me.

259. I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct.

Respectfully Submitted

/Trevor Mudge/

Trevor Mudge

Date: February 2, 2023