



Mercedes's Hearing Demonstratives for *Mercedes vs. Daedalus*

IPR2023-01335

U.S. Patent No. 8,775,833 B2

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

1

Mercedes EX1033
Mercedes-Benz USA, LLC
v. Daedalus Prime LLC
IPR2023-01335

Table of Contents



- 1** Overview of '833 Patent
- 2** Invalidity Overview
- 3** Dispute 1 – Whether Finkelstein Discloses Operation at an “Independent” Voltage/Frequency
- 4** Dispute 2 – Whether Finkelstein Discloses Dynamically Allocating a Power Budget of a Processor
- 5** Dispute 3 – Whether Nussbaum Teaches Per-Domain Power Budget Allocation According to a Preference Value
- 6** Dispute 4 – Whether Nussbaum + Bose Disclose Incrementing Power Sharing Values
- 7** Dispute 5 – Whether Conroy Teaches Different “Storage[s]” and “Current Time Interval”
- 8** Other Disputes

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

2

Overview of the '833 Patent

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

3

Mercedes EX1033
Mercedes-Benz USA, LLC
v. Daedalus Prime LLC
IPR2023-01335

Patent No. 8,775,833 B2 Overview



United States Patent
Ananthakrishnan et al.

(10) Patent No.: **US 8,775,833 B2**
(45) Date of Patent: ***Jul. 8, 2014**

(54) **DYNAMICALLY ALLOCATING A POWER BUDGET OVER MULTIPLE DOMAINS OF A PROCESSOR**

(71) Applicant: **Intel Corporation**, Santa Clara, CA (US)

(72) Inventors: **Ananthakrishnan, Arinath N.** (Hillsboro, OR (US)); **Efrain Rotem** (Haifa (IL)); **Doron Rajwan** (Rishon Le-Zion (IL)); **Elihu Weissmann** (Haifa (IL)); **Nadav Shulman** (Tel Mond (IL))

(73) Assignee: **Intel Corporation**, Santa Clara, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. (154b) by 0 days. This patent is subject to a terminal disclaimer.

(21) Appl. No.: **13/780,066**

(22) Filed: **Feb. 28, 2013**

(65) **Prior Publication Data**
US 2013/0179704 A1 Jul. 11, 2013

Related U.S. Application Data
Continuation of application No. 13/225,677, filed on Sep. 6, 2011.

(51) **Int. Cl.** (2006.01)
G06F 1/00 (2006.01)
G06F 15/00 (2006.01)
G06F 17/12 (2006.01)

(52) **U.S. Cl.** (2006.01)
USPC **713/900; 345/501; 700/286; 712/28**

(58) **Field of Classification Search**
USPC **713/900; 345/501; 700/286; 712/28**
See application file for complete search history.

(56) **References Cited**
U.S. PATENT DOCUMENTS
5,163,153 A 11/1992 Cole et al.
5,287,252 A 2/1994 Koenig et al.
5,522,087 A 5/1996 Huang
5,580,341 A 12/1996 Matter
5,621,250 A 4/1997 Kim
5,931,958 A 8/1999 Tisa
(Continued)

FOREIGN PATENT DOCUMENTS
EP 1,282,830 A1 2/2003
JP 08-020950 1/1997
JP 2000-202153 7/2001
(Continued)

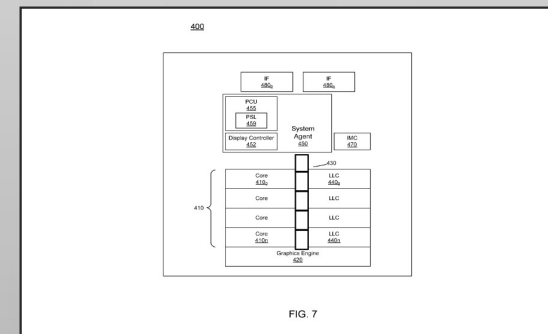
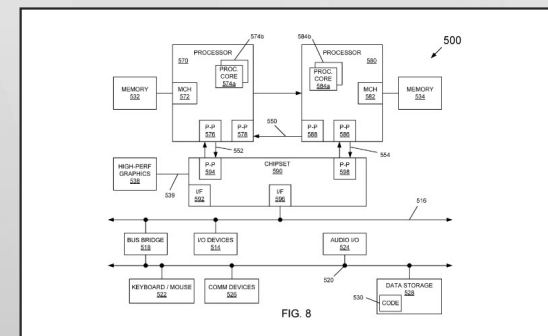
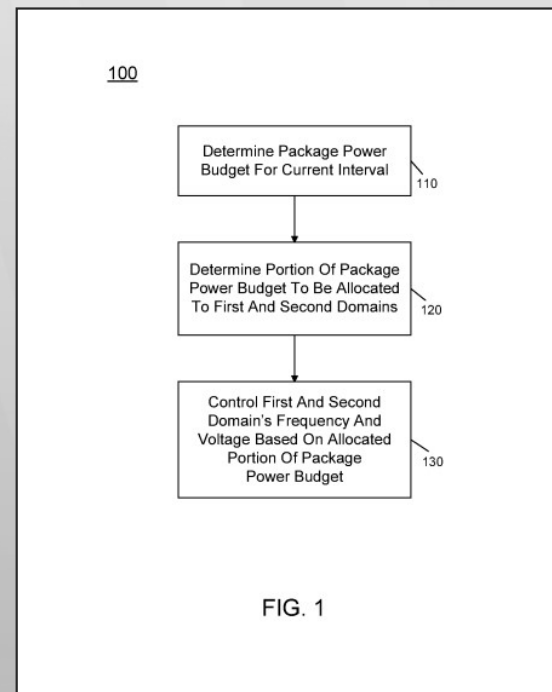
OTHER PUBLICATIONS
U.S. Appl. No. 13/262,896, filed Oct. 27, 2011, entitled "Enabling a Non-Core Domain to Control Memory Bandwidth," by Ananthakrishnan, et al.
(Continued)

Primary Examiner — **Stefan Stojanov**
(74) Attorney, Agent, or Firm — **Trop, Pruner & Hu, P.C.**

(57) **ABSTRACT**
In one embodiment, the present invention includes a method for determining a power budget for a multi-domain processor for a current time interval, determining a portion of the power budget to be allocated to first and second domains of the processor, and controlling a frequency of the domains based on the allocated portion. Such determinations and allocations can be dynamically performed during runtime of the processor. Other embodiments are described and claimed.

18 Claims, 8 Drawing Sheets

The '833 Patent relates to “determining a power budget for a multi-domain processor for a current time interval.”



Overview of the Claimed Invention – Independent Claim 1



1. A processor comprising:
a first domain and a second domain, each of the first and second domains to operate at an independent voltage and frequency;
a memory controller coupled to the first and second domains;
at least one interface; and
first logic to 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.

Ex-1001 at 11:26-38

Independent Claim 1 is representative of the alleged invention. It claims a processor with a:

- First domain and second domain operating at an independent voltage and frequency
- First logic to dynamically allocate a power budget for the processor
 - Between the first and second domains at run time
 - According to a sharing values for each domain (first and second)

Invalidity Grounds



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

Dispute 1 – Whether Finkelstein Discloses Operation at an “Independent” Voltage/Frequency

Claims 1-5, 7

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

7

U.S. Published Patent Appl. No. 2010/0115304 A1 (“Finkelstein”)

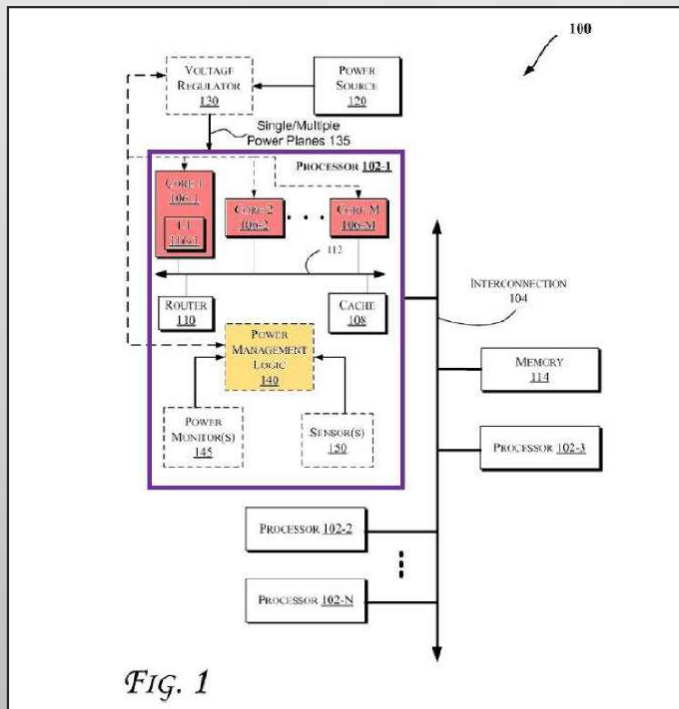


FIG. 1

Ex-1005 at Fig. 1 (Annotated)

- Just like the '833 Patent, Finkelstein “relates to power management for multiple processor cores”
- Finkelstein discloses a system that includes “one or more processors” where each processor “may include one or more processor cores”
 - Each processor may communicate with various other components
- Finkelstein 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

Whether Finkelstein Discloses “Independent” Voltage/Frequency



Claims 1-5 and 7 require “a first domain and a second domain, each of the first and second domains to operate at an **independent voltage and frequency.**”

- The first set of processor cores (red) is a “first domain” and another set of cores (blue) is a “second domain.”
- Finkelstein uses power management logic 140 (yellow) to control supply of power to components of the processor and may request the cores to modify their operating frequency, power consumption, etc.

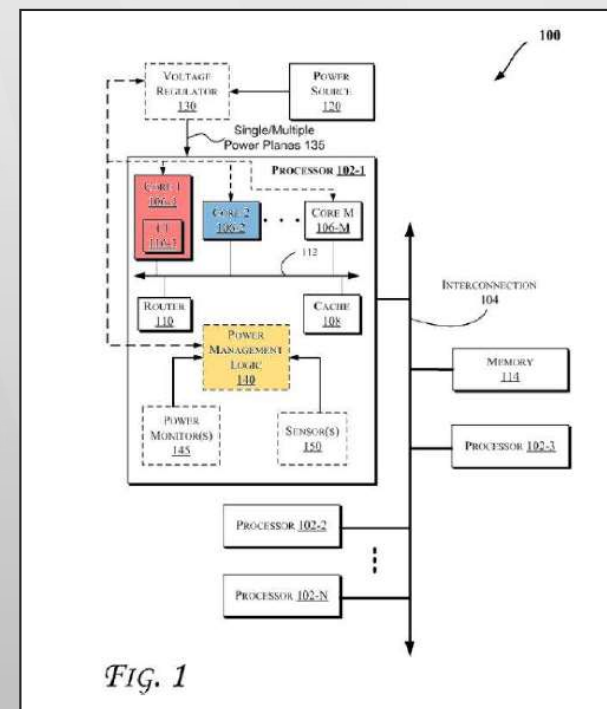


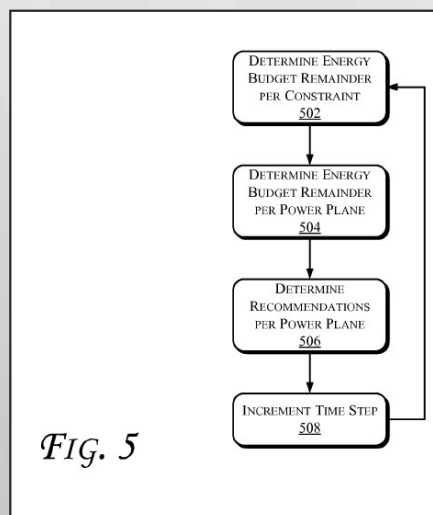
FIG. 1

Ex-1005 at Fig. 1 (annotated)

Whether Finkelstein Discloses “Independent” Voltage/Frequency



- Finkelstein discloses that the operation of these power planes are independent of each other, noting that control is “per power plane.”
- A POSITA implementing Finkelstein would understand that logic 140 could modify the voltage and/or frequency of either Core 1 (first domain) or Core 2 (second domain), **without affecting the operating voltage and/or frequency of the other.**



Ex-1005 at Fig. 5

[0027] Some current power-based management schemes may miss the temporal aspect of power management, since they generally address only the current point of time. Accordingly, in some embodiments, an energy budget **per power plane may** be used that permits: (1) to define both individual component (e.g., **individual processor core**) and shared constraints (e.g., shared amongst multiple components or processor cores) in a way that takes temporal aspects into account; (2) to express effective constraints corresponding to different time constants; (3) to handle the case of on-line changes in the constraints. In an embodiment, power distribution may be managed among different power planes and under energy-based definition(s).

Ex-1005 at ¶27

Whether Finkelstein Discloses “Independent” Voltage/Frequency



Patent Owner argues:

- “Independent” is used to mean “separately controlled;” and
- An “independent” first and second domain “does not preclude the voltage and frequency of the first domain from having a relationship to, or dependency on, the voltage and frequency of the second domain.”

But a POSITA would know:

- “Separately controlled” does not appear in the text of the patent
- The “separately controlled” definition includes a self-contradictory assertion that “independent” can in fact mean “dependent”
- In fact, Petitioner and Patent Owner agree that independence still allows for the possibility that the voltage and frequency can, at some times, be the same across domains.

Whether Finkelstein Discloses “Independent” Voltage/Frequency



Patent Owner argues:

- “Independent” is used to mean “separately controlled;” and
- An “independent” first and second domain “does not preclude the voltage and frequency of the first domain from having a relationship to, or dependency on, the voltage and frequency of the second domain.”

But a POSITA would know:

- Patent Owner’s expert similarly does not use “separately controlled” when describing claim 1:

Q. Okay. And then, getting into the claims, for instance, in Claim 1, it's noting that the first and second domain have operated independent voltage and frequency.

Do you agree with that?

A. That the -- the claim says the first -- sorry -- a first domain and a second domain, each of the first and second domains, to operate at an independent voltage and frequency, is talking about these domains -- when we look at the equations or the formulas -- and as I talk about in my Declaration, they are related, but each of the values or budgets for the domains can be -- can be set or operated independently.

Brogioli Depo. (Ex-1032) at 24:12-24

Whether Finkelstein Discloses “Independent” Voltage/Frequency



In the context of the claims, a POSITA would understand “independent” means “each domain **could have** a different voltage and frequency than the other domain”

Patent Owner’s expert agrees:

Q. So each domain could have a different voltage than frequency than the other domain, not necessarily saying that it would, but it could; true?

A. I think that sounds right.

Brogioli Depo. (Ex-1032) at 28:7-10

Dispute 2 – Whether Finkelstein Discloses Dynamically Allocating a Power Budget of a Processor

Claims 1-5, 7

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

14

Whether Finkelstein Discloses Dynamically Allocating a Power Budget



- Finkelstein discloses where multiple power planes share a power source, a shared common package power/energy budget is deployed to allow power to be shifted from one plane to another
- Finkelstein provides an example of shifting unused processor core power to a graphics processor when graphics effects are exhibiting a more intensive workload and processor resources are not being fully utilized.
- A POSITA would have understood this is a **runtime** shift based on workloads, and as such, dynamically allocating the power budget.

[0026] Additional power management challenges may however be present with respect to processors having multiple power planes. For example, if multiple power planes sharing the same power source **120** are present in system **100**, power management may need to be satisfied for both individual constraints per power plane and global constraints per package. By doing so, it is possible to share a common package power/energy budget and allow for use of, for example, unused processor core power for more Graphics Effect(s) (GFX) performance in a GFX intensive workload, when processor resources are not fully utilized. However, the techniques discussed herein with reference to multiple power planes may also be applied to implementations that may utilize a single power plane.

Ex-1005 at ¶ 26

Whether Finkelstein Discloses Dynamically Allocating a Power Budget



- Finkelstein discloses an energy budget E^k where k represents a constraint – a preference that describes how the budget is distributed among power planes.
- W_k is a set of weighting values applied to E^k such that W_{ki} is a scaling of E^k for a particular domain i , and $W_{ki}E^k$ is the portion of the energy budget attributable to that domain i under the constraint k .

$$E^{ki} = W_{ki}E^k$$

- With that understanding, Finkelstein discloses that the “unused” budget for domain i is defined by

$$W_{ki}E^k - E_{high}^i$$

Finkelstein discloses that this unused budget may be “distributed among the rest of the power planes proportionally to their weights.”

[0034] Regarding the constraints, assume existence of m controllers, $f^i(E)$ corresponding to different power planes. For constraint k , user (or application defined) preferences that describe how budget E^k is distributed among power planes may be determined. For example, 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 . For individual constraints, a single power plane may obtain the entire budget, so the corresponding weight vector is a unit vector. In the general case, power plane i may obtain the portion of the budget:

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

[0035] In some cases, we may have that for a power plane i , its portion $W_{ki}E^k$ is high enough to provide maximal turbo (namely, $W_{ki}E^k \geq E_{high}^i$). In this case, the “unused” budget for this power plane, $W_{ki}E^k - E_{high}^i$ may be distributed among the rest of the power planes proportionally to their weights.

Ex-1005 at ¶ 34-35

Whether Finkelstein Discloses Dynamically Allocating a Power Budget



Patent Owner argues:

“[T]he ‘unused’ budget for domain i ($W_{ki}E^k - E_{high}^i$) is **not** a power budget **for a processor**.”

Instead, any “unused” budget “is an excess of a power budget of a power plane.”

But a POSITA would understand:

- The claim requires “dynamically allocat[ing] a power budget for the processor **between the first and second domains**”
- Finkelstein discloses that the total power budget is E^k , which is distributed among domains i based on domain weights W_{ki} .
- Finkelstein further discloses that “if multiple power planes [are] sharing the same power source,” then these planes may “share a common package power/energy budget.”
- Thus, a POSITA would understand Finkelstein to disclose dynamically allocating a power budget of **the processor between the first and second domains**.

Whether Finkelstein Discloses Dynamically Allocating a Power Budget



- Finkelstein (like the '833 patent) is concerned with dynamically allocating power for “**processors** having multiple power planes” which share the same power source and “a common **package** power/energy budget”
- Patent Owner's expert confirmed that a package is “the whole”, *i.e.*, the entire processor.
- Thus, Finkelstein's disclosure of sharing common package power/energy budget is allocating a power budget for the processor between the first and second domains.

Patent Owner's expert states:

Q. What's a package in this context?

A. It's the whole -- the cores and other things that may be with them on the chip and, sort of, packaged in the -- what we kind of see in pictures like the black rectangle with the metal things coming out.

Brogioli Depo. (Ex-1032) at 38:1-5

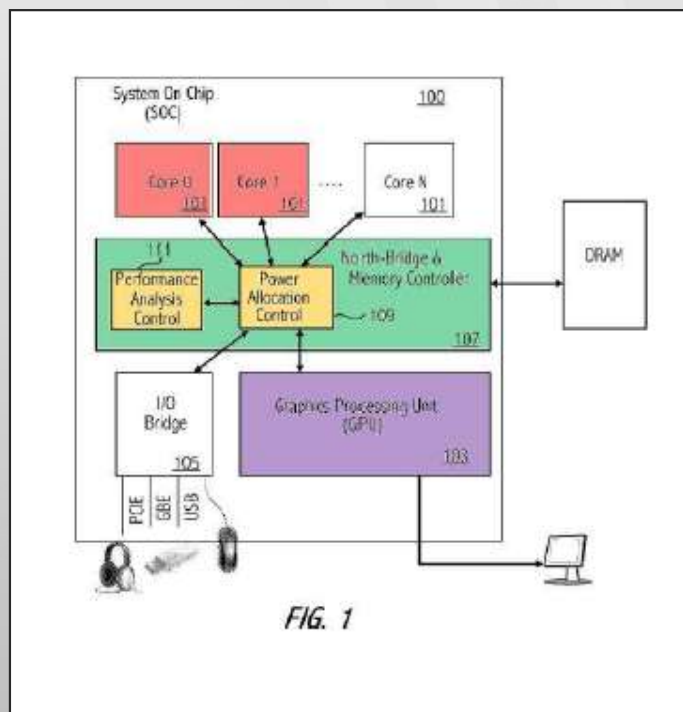
Dispute 3 – Whether Nussbaum Teaches Per-Domain Power Budget Allocation According to a Preference Value

Claims 13-15, 17-18

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

19

U.S. Published Patent Appl. No. 2011/0022356 A1 (“Nussbaum”)



Ex-1007 at Fig 1 (Annotated)

- Nussbaum discloses power management of computational units and allocation of “power headroom” to various on-die or on-platform components
- Power budget is dynamically allocated according to whether an application is “CPU-bounded or GPU-bounded”
- Nussbaum describes the reallocation of power from less active components to busy ones
- Nussbaum explains that boost sensitivity—“performance sensitivity of each computational unit to frequency change”—is determined and stored on a computational unit basis

Whether Nussbaum Teaches Per-Domain Power Budget Allocation

13. A system comprising:

a multicore processor having **a first domain including a plurality of cores**,
a second domain including a graphics engine, and
a third domain including system agent circuitry,
the third domain to operate at a fixed power budget and including 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; and
a dynamic random access memory (DRAM) coupled to the multicore processor.

Whether Nussbaum Teaches Per-Domain Power Budget Allocation



Patent Owner argues:

“[I]n Nussbaum, frequency sensitivities are per-component, e.g., per-core, attributes, not per-domain attributes.”

But a POSITA would understand:

- The **first domain including a plurality of cores** is Nussbaum’s Core 0 and Core 1.
- Patent Owner assumes that the term “computational unit” in Nussbaum can only refer to a “core.”
- But Nussbaum repeatedly refers to “cores or other computational units”
- Further, Nussbaum provides examples of “computational units,” which includes CPUs or graphical processing units (GPUs).

Whether Nussbaum Teaches Per-Domain Power Budget Allocation



- Nussbaum discloses distributing power headroom between domains
- A first domain is represented by Core 0 and Core 1
 - Nussbaum discloses that the cores are equally boosted to a higher F, V to fill new power headroom.
- A second domain is represented by the Graphics Processing Unit
- A third domain is represented by Core 2 and Core 3.
- Nussbaum discloses distributing power from the Core 2/Core 3 domain to the Core 0/Core 1 domain.

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	

Ex-1007 at Tables 1, 2

Whether Nussbaum Teaches Per-Domain Power Budget Allocation



Patent Owner argues:

“[I]n Nussbaum, frequency sensitivities are per-component, e.g., per-core, attributes, **not** per-domain attributes.”

“[E]ach one of the cores in such a domain will, necessarily, have its own frequency sensitivity.”

But a POSITA would understand:

- A “computational unit” can include more than one core, in other words, a “computational unit” can be a “domain.”
- Frequency sensitivities are on a computational unit basis (e.g., a domain basis).
- Nussbaum “**distribut[es]**” “power headroom **among the computational units**” and allocates power between them.

[0019] The above approaches for homogenously increasing power to all cores or to one or more cores based on performance states of the cores, allow for power to be reallocated from idle computational units, such as the CPU or graphical processing unit (GPU), but treat all active units homogeneously when dithering frequency or boosting steady state frequency. However, some active cores or other computational units may be gaining little or no performance increase from a higher core frequency, while other cores or computational units may be running workloads with a higher sensitivity to an increase in core frequency. Selectively distributing power among the active cores or other computational units based on frequency sensitivity allows for greater overall system throughput on heterogeneous workloads or multi-threaded workloads with heterogeneous threads. That requires an effective approach to identify workload sensitivity to changes in core frequency.

Ex-1007 at [0019]

Whether Nussbaum Teaches Per-Domain Power Budget Allocation



Patent Owner argues:

"[I]n Nussbaum, frequency sensitivities are per-component, e.g., per-core, attributes, **not** per-domain attributes."

"[E]ach one of the cores in such a domain will, necessarily, have its own frequency sensitivity."

But a POSITA would understand:

- Nussbaum provides:

[0023] An embodiment provides reallocation of the power from idle or less active components to the busy components by having more power allocated to the busy ones. For

[0025] According to an embodiment, one way to determine how to allocate power between Core0 and Core1 to try and achieve improved performance gain 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. Changes in performance capability may also be provided by, e.g., a change in the amount of cache available to a core, the number of pipelines operating in the core and/or the instruction fetch rate. In order to evaluate which of the cores can better exploit an increase in performance capability, in one embodiment, 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.

Ex-1007 at [0023], [0025]

Whether Nussbaum Teaches “Minimum Reservation Value[s]”



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.

Whether Nussbaum Teaches “Minimum Reservation Value[s]”



According to the '833 Patent, the “minimum reservation value” is the minimum amount of power budget that can be allocated to a domain.

Referring now to FIG. 2, shown is a flow diagram of a method describing further details of allocating a package power budget between multiple domains in accordance with an embodiment of the present invention. As shown in FIG. 2, method 200 may similarly be performed by power sharing logic of a PCU or other power controller of a processor. Method 200 can begin by obtaining minimum reservation values for the domains (block 210). As in the embodiment of FIG. 1 assume a multi-domain processor including at least a core domain and a graphics domain. These minimum reservation values can be obtained, e.g., from configuration registers set by user-level software to indicate a floor level corresponding to a minimum amount of power budget to be allocated to the given domain. Of course, these values instead can be set by other entities such as an OS and/or graphics driver, respectively for the core and graphics domains.

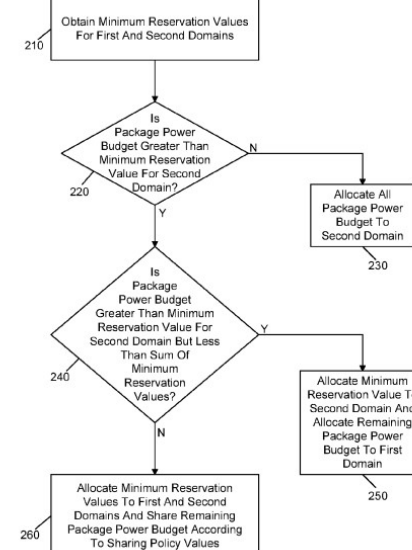


FIG. 2

Ex-1001 at 5:65-6:13, Figure 2



Whether Nussbaum Teaches “Minimum Reservation Value[s]”

- “**Deep Cstate**” and “**P3**” represent “minimum reservation value[s]” for the first domain.
- “**GPU_P3**” represents a second minimum reservation value for the second domain.
- A POSITA would understand that the lowest operational points of Table 3 and 4 in Nussbaum correspond to lower performance states in each respective domain

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

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

Ex-1007 at Tables 3, 4

Whether Nussbaum Teaches “Minimum Reservation Value[s]”



Patent Owner argues:

“Nussbaum’s Deep Cstate is characterized by ‘Clocks Off/Power Off.’”

“[Deep Cstate] is instead a state when one can expect no power to be allocated.”

But a POSITA would understand:

- Nussbaum discloses “*Power . . . consumed*” at the “Deep Cstate” power setting as “Core_DeepCstate_Pwr.”
- A “Deep Cstate” core can receive its **minimum reservation value** of power even in this state.

Whether Nussbaum Teaches “Minimum Reservation Value[s]”



Patent Owner does not dispute that a “**Deep Cstate**” core can receive its “minimum reservation value” of power even when in the “**Off/Power Off**” state of Table 3.

Patent Owner’s expert agrees:

Q. What is deep voltage?

A.

And, here, it looks like it's either the power has been gated to it, or there's a very, very deep -- perhaps extremely low voltage applied to maybe maintain some kind of -- something internal, but there's no functionality going on, no computation, et cetera.

Q. So for the deep C-state, there might be some, maybe, tiny modicum of voltage that's still being applied, correct? That's what you just testified to, extremely low voltage applied?

A. I think that seems to be what it's -- the remarks that Table 3 are getting at, that there's -- there may be some extremely low voltage applied.

Brogioli Depo. (Ex-1032) at 64:1-19

See IPR2023-01335 PO Response at 50-52; PO Sur-reply at 15-16.
IPR2023-01335 Pet. at 48-50; Pet. Reply at 11-12.

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

30

Whether Nussbaum Teaches “Minimum Reservation Value[s]”



Patent Owner argues:

“[T]o the extent power is allocated to cores or GPUs in accordance with performance states [(P3 and GPU_P3)], those allocations are made on a per-computational unit basis.”

“These are not domain-based allocations as specified in claim 17.”

But as established earlier:

- A “computational unit” can include more than one core, in other words, a “computational unit” can be a “domain.”
- Nussbaum “**distribut[es]**” “power headroom **among the computational units**” and allocates power between them.

Whether Nussbaum discloses a “Preference Value”



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.

Whether Nussbaum discloses a “Preference Value”



According to the '833 Patent, the “preference value” is used to favor a specific domain for power allocation based on, for example, workload of a processor.

Thus the embodiment shown in FIG. 2 favors providing power to the second domain (which may correspond to a graphics domain) over the first domain (which may correspond to a core domain). Although shown with this preference in FIG. 2, understand the scope of the present invention is not limited in this regard, and in other implementations the preference can be in the other direction. And note that this preference can be dynamically changed based on a workload of a processor (e.g., a graphics intensive workload versus a computing intensive workload). As an example, a preference for the graphics domain can be hard mapped to first allocate to the graphics domain and then to the core domain to share the rest between core domain and graphics domain. This decision can be made based on the POLICY_FIRST and POLICY_SECOND values. For example, if POLICY_FIRST is greater than POLICY_SECOND, the preference may be to first allocate to the core domain and then allocate the remaining budget to graphics domain and so on.

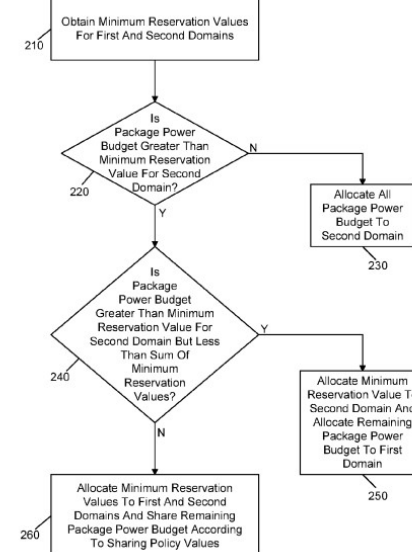


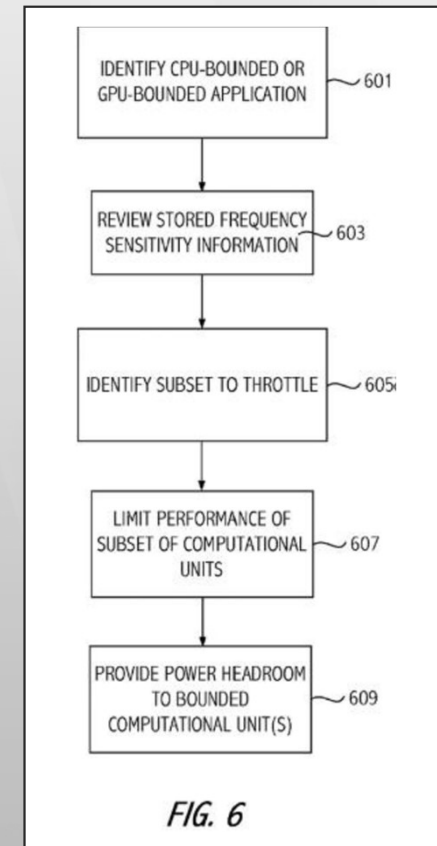
FIG. 2

Ex-1001 at 5:65-6:31, Figure 2

Whether Nussbaum discloses a “Preference Value”



- Step 601 identifies an application as “GPU-bounded” or “CPU-bounded.”
- Nussbaum explains that if an application is GPU-bounded, then the CPU cores may be throttled to release power margin available to the GPU.
- If the GPU is the second domain and the CPU is first domain, then a POSITA would understand that an output value of step 601 is a “**preference value**” to favor the second domain over the first domain.
- “As shown in Figure 6, Nussbaum discloses allocating its variable power budget based on an output value of step 601. . .”
 - Step 609 (shown in Figure 6) discloses allocating based on step 601 preference value.



Ex-1007 at Fig. 6

Whether Nussbaum discloses a “Preference Value”



Patent Owner argues:

“The ‘output value’ of step 601 is not used to ‘allocate the variable power budget,’ but rather to adjust the amount of power budget that is available.”

But a POSITA would understand:

The steps provided in Figure 6 of Nussbaum allocate power budget according to the preference value of Step 601.

Patent Owner expert concedes:

Q. Okay. So maybe not just 601, but maybe 601 through 609, the performance of those five steps determines whether the GPU or CPU will be selected for additional power headroom to be provided, correct?

A. The process and flow described in Figure 6 is being used to identify those two throttles based on the subset of cores being less sensitive in terms of performance or reduction performance capability. So it's talking about the throttling here.

Brogioli Depo. (Ex-1032) at 66:19-67:2

Dispute 4 – Whether Nussbaum + Bose Disclose Incrementing Power Sharing Values

Claim 16

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

36

Whether Nussbaum + Bose Disclose Incrementing Power Sharing Values



Claim 16 provides: 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 a second domain is not granted**.

Whether Nussbaum + Bose Disclose Incrementing Power Sharing Values



Nussbaum discloses:

- “[W]hen the context changes, the sensitivity is re-evaluated.”
- Boost sensitivity is adjusted (i.e., incremented) to components which “are the most sensitive to boosting and thus ‘deserve’ to get a higher TDP power margin reallocated when a reallocation takes place.”

A POSITA would understand:

Nussbaum explicitly contemplates incrementing boost sensitivities of computational units, including computational units whose requests for a higher frequency were not granted.

- Boost sensitivities represents **deserving** more power that hasn't yet been granted.

160. 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 (Nussbaum) ¶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.

Horst Decl. (Ex-1029) at ¶ 160

[0007] In another embodiment, an integrated circuit is provided that includes a plurality of computational units and control logic configured to determine respective performance sensitivities for each of the plurality of computational units. In an embodiment, the control logic applies clock signals at respective first frequencies to respective ones of the computational units during respective first evaluation periods and determines respective first utilization metrics. The control logic applies the respective clock signals at respective second frequencies to the respective ones of the computational units during respective second evaluation periods and determines respective second utilization metrics. The respective performance sensitivities for the plurality of computational units may be determined based on the respective first and second utilization metrics. In an embodiment, the integrated circuit determines **boost sensitivity of a computational unit in response to a process context change associated with the computational unit.** The integrated circuit may determine performance sensitivity of a computational unit in response to passage of a predetermined period of time since a performance sensitivity was previously determined for the computational unit.

Ex-1007 at [0007]

[0032] The Boost Sensitivity Table (BST) is maintained as a result of a frequency sensitivity training session for the components to be potentially boosted. In other embodiments, a frequency sensitivity table is maintained as a result of the frequency sensitivity training for all components whose performance can be adjusted, typically through adjusting frequency (and voltage if necessary). In an embodiment, power budget reallocation uses the information in the BST to decide which on-die component(s) are the most sensitive to boosting and thus “deserve” to get a higher TDP power margin reallocated when a reallocation takes place.

Ex-1007 at [0032]

Whether Nussbaum + Bose Disclose Incrementing Power Sharing Values



Patent Owner argues:

“the ‘boost values’ are physical characteristics calculated based on performance level and utilization metrics, and that while they are periodically re-evaluated, they are not ‘incremented’ in response to a request.”

“Nussbaum describes re-evaluating the boost sensitivities, it does not describe intentionally ‘increment[ing]’ them (i.e., raising the value) as in the ‘833 patent.”

But a POSITA would understand:

Nussbaum explicitly contemplates incrementing boost sensitivities of computational units, initiated by a timer or context switch, to adjust boost sensitivity for components that “‘deserve’ to get a higher TDP power margin.”

Nussbaum explicitly contemplates raising the value during re-evaluation:

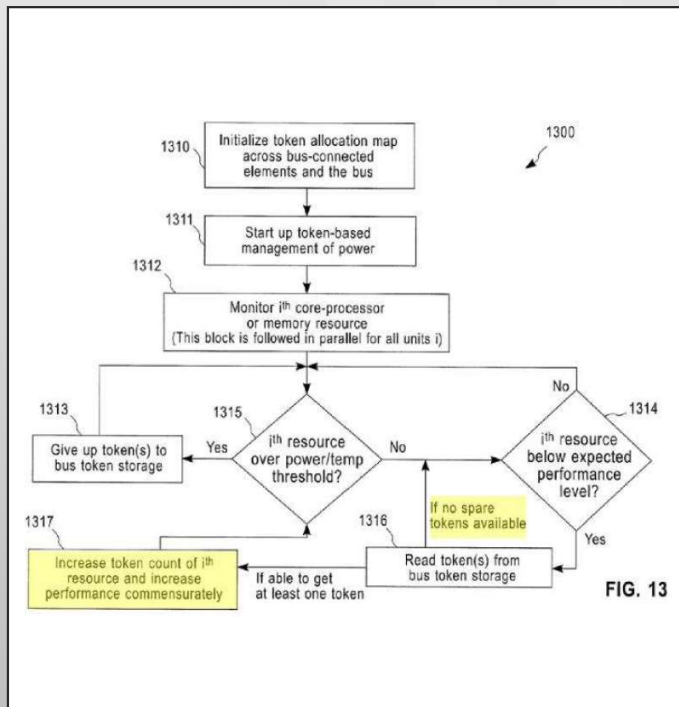
Id. ¶30. Nussbaum also explains that boost sensitivity may “expire[] for each context based on a fixed or programmable timer,” and in some implementations, “both a timer and context switch, whichever occurs first, are used to initiate the boost sensitivity reevaluation.” *Id.* Thus, a POSITA would have 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.

Horst Decl. (Ex-1029) at ¶ 160

[0032] The Boost Sensitivity Table (BST) is maintained as a result of a frequency sensitivity training session for the components to be potentially boosted. In other embodiments, a frequency sensitivity table is maintained as a result of the frequency sensitivity training for all components whose performance can be adjusted, typically through adjusting frequency (and voltage if necessary). In an embodiment, power budget reallocation uses the information in the BST to decide which on-die component(s) are the most sensitive to boosting and thus “deserve” to get a higher TDP power margin reallocated when a reallocation takes place.

Ex-1007 at [0032]

U.S. Published Patent Appl. No. 2009/0089602 A1 (“Bose”)



Ex-1008 at Fig 13 (Annotated)

- Bose discloses power management of a system of connected components
- Each component is assigned a power budget as determined by a number of allocated tokens
- Each token connotes a specified quantum of power, where total tokens determines total power limit
- 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.

Whether Nussbaum + Bose Disclose Incrementing Power Sharing Values



- Patent Owner does not dispute that step 1316 is a request for additional tokens that, if denied, triggers a loop leading back to step 1316.
- When a component in Bose requests an increase in power, Bose responds by attempting to increase the token count (the new “power sharing value”) of the component.

Nussbaum Recites:

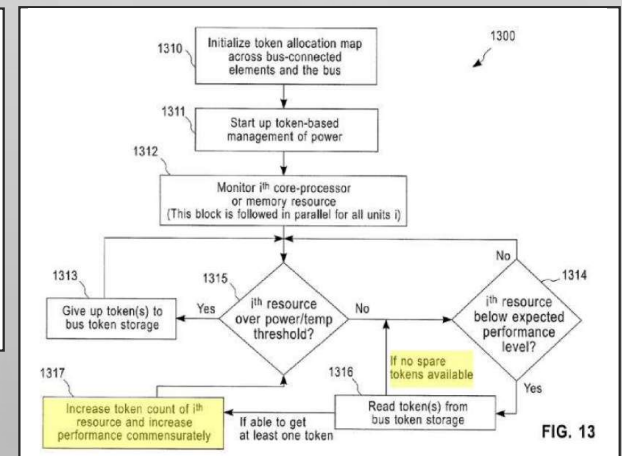
[0030] The boost sensitivity for each core is tied to the current processor context, which can be approximated by the x86 register value of CR3, tracked by the North-Bridge. In one embodiment, when the context changes, the sensitivity is re-evaluated. In another embodiment, the boost sensitivity expires for each context based on a fixed or programmable timer (e.g., after 1-100 ms). In still other embodiments, both a timer and context switch, whichever occurs first, are used to initiate the boost sensitivity reevaluation.

Ex-1007 at [0030]

Bose Recites:

[0086] If a monitored unit is assessed to be above a predefined power or temperature threshold (in block 1315), it is forced to give up one or more tokens to the centralized bus token storage in block 1313, until the power/thermal overrun is mitigated. Alternatively, if the resource is below power and temperature limits, its performance level is gauged in block 1314. If it is assessed to be below a predefined minimum performance, block 1316 is used to try and read additional tokens from the centralized bus token storage; such extra tokens are used to boost up performance in the targeted unit in block 1317 (e.g., by increasing its voltage and frequency, if allowed by the system architecture) and then block 1315 is again entered to check if the resource is within the power and temperature limits.

Ex-1008 at [0086], Fig. 13



DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Whether Nussbaum May Be Combined with Bose



Patent Owner argues:

“[T]here is no reason for a person of ordinary skill in the art to adopt a scheme such as in Bose for incrementing a token count to determine when to satisfy a request inasmuch as there is no such request to be satisfied in Nussbaum.”

“Petitioner does not identify any teaching or suggestion that would have provided that alleged motivation in the art.”

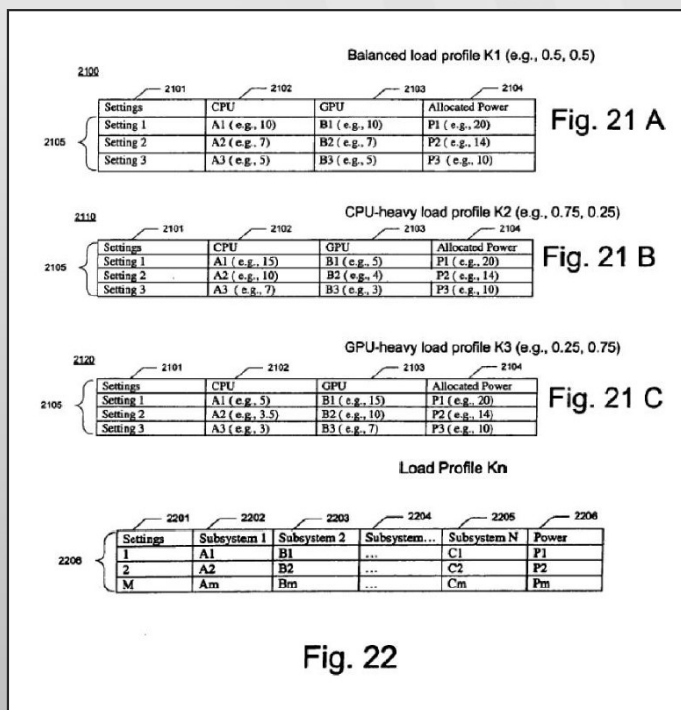
But a POSITA would understand:

- The need to increase the likelihood that a core in need of a higher frequency would be “boosted.”
- Nussbaum prefers boosting “cores with the highest [boost] sensitivity,” and contemplates incrementing boost sensitivities of computational units.
- Bose discloses a power management system with a “token management unit (TMU)” to “try and read additional tokens from [a] centralized bus token storage ... to boost up performance.”
- Nussbaum’s power allocation operations, with Bose’s teachings, increment a boost sensitivity of a particular domain.

Dispute 5 – Whether Conroy Teaches Different “Storage[s]” and “Current Time Interval”

Claim 13-15, 17-18

U.S. Published Patent Appl. No. 2007/0049133 A1 (“Conroy”)



Ex-1006 at Fig 22

- Conroy is titled “Methods and apparatuses for dynamic power control”
- Conroy discloses “dynamically redistribut[ing] power in a [computer] system that includes a plurality of subsystems.”
 - A “subsystems may be, for example, a central processing unit (‘CPU’) and a graphics processor (‘GPU’).”
- Conroy’s power distribution table may correspond to various power distribution profiles, including where one subsystem requires most of the power followed by another subsystem requiring most of the power

Whether Conroy Teaches Different “Storage[s]”



13. A system comprising:

a multicore processor having a first domain including a plurality of cores,
a second domain including a graphics engine, and
a third domain including system agent circuitry,

the third domain to operate at a fixed power budget and

including 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**; and

a dynamic random access memory (DRAM) coupled to the multicore processor.

Whether Conroy Teaches Different “Storage[s]”



Conroy uses a memory to store a table of power sharing values, and Conroy’s individual “power sharing value[s]” would be contained in separate registers.

Increasing in a Target Goal Level

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

Ex. 1006 at Fig. 7

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

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

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

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

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

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

Ex. 1006 at Fig. 21

Whether Conroy Teaches Different “Storage[s]”



Patent Owner argues:

Despite the claim language requiring a first and second storage, “Conroy...makes clear that all of the power distribution tables, and, therefore, all of the contents of each table, are stored in a memory—that is, one memory.”

But a POSITA would understand:

Conroy discloses storing power sharing values (“throttle settings” i.e., voltage and/or frequency settings) in registers.

Registers are storage.

[0097] For example, the microcontroller may store the determined throttle setting in a register and then send a signal to the corresponding component (or the main CPU) to enforce the throttle setting. In one embodiment, the microcontroller sends the signal to enforce the throttle change only when the dynamically determined throttle setting is different from the previous one.

One or more measuring devices 3707, e.g., one or more sensors as described above with respect to FIGS. 4, 5, 12, 17, 24, and 32 are coupled to subsystems 3701-3705, and to power manager 3708, as shown in FIG. 37. A power look-up table 3709 that may include a power distribution table, as described above with respect to FIGS. 7, 21, and 22, is coupled to power manager 3708, as shown in FIG. 37. Components of the system 3700, including processors, microcontrollers, buses, I/O controllers, I/O devices, memories, sensors are described in detail above with respect to FIGS. 1-17. In one embodiment, one or more power lookup tables corresponding to various performance settings of the computer system as described above with respect to FIGS. 1-36, may be generated by subsystem 3701 (or generated by test equipment in the design and/or manufacturing process), and stored in memory 3704, and/or in a memory located in power manager 3708. In one embodiment, power manager 3708 performs methods described above with respect to FIGS. 33-36. In another embodiment, subsystem 3701 performs methods described above with respect to FIGS. 33-36.

Ex-1006 at [0097], [0225]

Whether Conroy Teaches a “Current Time Interval”



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.

Whether Conroy Teaches a “Current Time Interval”



[0069] In FIG. 2, actual power usages are monitored. For example, the actual power usage can be measured periodically to determine the history of the power usage. The history of the power usage can be used to determine the power usage in certain averaged ways.

[0070] In one embodiment of the present invention, with the knowledge of the past power usage (e.g., measurements 211, 213, . . . , 219 at times $T-(n-1)\Delta$, $T-(n-2)\Delta$, . . . , T) the system can dynamically determine the allowable power budget for the next time interval (e.g., for time $T+\Delta$).

Ex-1006 at [0069]-[0070]

- Conroy discloses an iterative process with each time interval described as “past” or “next”
- “Next” means that which comes after the past, *i.e.* the present
- “T” is associated with the sequence of time intervals in the “past time interval,” while “ $T+\Delta$ ” is associated with the “next time interval.”

Whether Conroy Teaches a “Current Time Interval”



Patent Owner argues:

“ $T+\Delta$ ” is a “next” time interval, and **not** the current time interval.

A POSITA would understand:

$T+\Delta$ is the “next time interval” after “past,” which is the “current time interval.”

191. 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 $T-(n-1)A$, $T-(n-2)A$, ... , T)” to “*dynamically determine the allowable power budget*” for time interval $T+A$, and allocates this “dynamically determined” power budget to different components of the system. Ex-1006 (Conroy) ¶¶12, 70-72, 80-83, 90, 120. A POSITA would have understood time interval $T+A$ to be a “current time interval” because Conroy refers to the time intervals preceding $T+A$ as being in the “*past*.” *Id.* ¶¶70-72, Fig. 2.

Horst Decl. (Ex-1029) at ¶ 191

Other Disputes

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

51

Mercedes EX1033
Mercedes-Benz USA, LLC
v. Daedalus Prime LLC
IPR2023-01335

Whether a POSITA Would Have Combined Conroy and Bose

Claim 16

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

52

Whether a POSITA Would Have Combined Conroy and Bose



- 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 iteration.
- Adding Bose's token incrementing scheme to Conroy's power manager would keep the power supplied to the various subsystems stable within each iteration.
- Conroy's paragraph 130 maintains selected throttle setting until the "next iteration" of dynamic throttling, just as Bose maintains a token count until the "next iteration" of its process.

Whether a POSITA Would Have Combined Conroy and Bose



Patent Owner argues:

- “[I]n Conroy, subsystems do not request increases in system settings.”
- “A person of ordinary skill in the art would have recognized that Conroy’s method of power budgeting does not rely on subsystems making ‘requests’ for power.”

But Conroy discloses such methods itself:

[0130] Typically, 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.

[0131] Alternatively, the throttle setting may be determined on a substantially continuous basis; and the power management system requests throttle setting changes when necessary. To avoid frequent changes in throttle settings, the power management system may determine the throttle setting so that the throttle setting will be valid for at least a predetermined period of time unless a significant change in the estimate of the power used by the non-throttled components is detected (e.g., when a disk is loaded into the DVD ROM drive).

Ex-1006 at [0130-31]

Whether a POSITA Would Have Combined Conroy and Bose



Patent Owner Argues:

In Conroy, “in all cases, it is the predicted power needs, not a request for power, that is the basis for changing power settings.”

A POSITA would understand:

In Figs. 21-22, a throttle setting change in Conroy corresponds to a change in “allocated power,” when required.

204. 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 (Conroy) ¶204.

Load Profile Kn

2200

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

Ex-1006 (Conroy), Fig. 22.

Horst Decl. (Ex-1029) at ¶ 204

Whether a POSITA Would Have Combined Conroy and Finkelstein

Claims 1-5, 7, 13-15, and 17-18

Whether a POSITA Would Have Combined Conroy and Finkelstein

- Conroy discloses “dynamically redistribut[ing] power in a system that includes a plurality of subsystems.”
 - 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.”
- Finkelstein discloses the distribution of a power budget and sharing an “energy budget” and explains that “higher power draw is allocated to and allowed for the subsystem having a higher workload.”

Whether a POSITA Would Have Combined Conroy and Finkelstein



A POSITA would be motivated to combine the teachings of Conroy and Finkelstein and 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.

Whether a POSITA Would Have Combined Conroy and Finkelstein



Patent Owner argues:

Petitioner's "only justification provided for [the] contention" that a POSITA would be motivated to implement Conroy's system on a chip "is that integrated circuits can accommodate 'additional functionality.'"

But a POSITA would understand:

implementing Conroy on a chip would (1) reduce size and weight of a system, (2) improve reliability, and (3) improve power efficiency.

212. 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 (Finkelstein) ¶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.

Horst Decl. (Ex-1029) at ¶ 212

Whether a POSITA Would Have Combined Conroy and Finkelstein



Patent Owner argues:

- “[H]ere, the teachings of Conroy and Finkelstein, are directed to different levels within a system.”
- “[M]odifying Conroy with Finkelstein would merely add another (Finkelstein-like) component within Conroy’s system.”

But a POSITA would understand:

The overlap between Conroy and Finkelstein enhances the amenability of a predictable combination.

212. 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 (Finkelstein) ¶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.

Horst Decl. (Ex-1029) at ¶ 212