

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MERCEDES-BENZ USA, LLC,

Petitioner,

v.

DAEDALUS PRIME LLC,

Patent Owner.

Case No.: IPR2023-01335

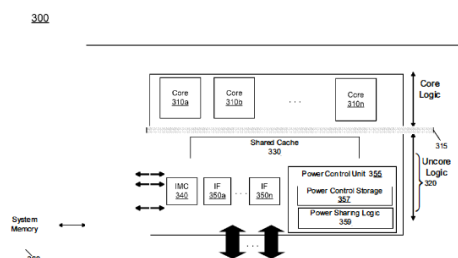
U.S. Patent No. 8,775,833

Patent Owner's Demonstrative Exhibits

MB USA v. Daedalus Prime IPR2023-01335
Patent Owner's Demonstrative Exhibits -- Not Evidence
Daedalus Ex. 2006

'833 Patent

- (12) **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:** Avinash N. Ananthakrishnan, Hillsboro, OR (US); Efraim Rotem, Haifa (IL); Doron Rajwan, Rishon Le-Zion (IL); Eliezer 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. 154(b) by 0 days.
This patent is subject to a terminal disclaimer.
- (21) **Appl. No.:** 13/780,666
- (22) **Filed:** Feb. 28, 2013
- (65) **Prior Publication Data**
US 2013/0179704 A1 Jul. 11, 2013
- Related U.S. Application Data**
- (63) Continuation of application No. 13/225,677, filed on Sep. 6, 2011.
- (51) **Int. Cl.**
G06F 1/00 (2006.01)
G06F 15/00 (2006.01)
G05D 3/12 (2006.01)
- (52) **U.S. CL**
USPC 713/300; 345/501; 700/286; 712/28
- (58) **Field of Classification Search**
USPC 713/300; 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.
5,287,292 A 2/1994 Kenney et al.
5,522,087 A 5/1996 Hsiao
5,590,341 A 12/1996 Matter
5,621,250 A 4/1997 Kim
5,931,950 A 8/1999 Hsu
(Continued)
FOREIGN PATENT DOCUMENTS
EP 1 282 030 A1 2/2003
JP 09-026900 1/1997
JP 2001-202155 7/2001
OTHER PUBLICATIONS
U.S. Appl. No. 13/282,896, filed Oct. 27, 2011, entitled "Enabling a Non-Core Domain to Control Memory Bandwidth," by Avinash N. Ananthakrishnan, et al.
(Continued)
Primary Examiner — Stefan Stoyanov
(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 portions. 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 proposes an approach for managing power consumption within integrated circuits, such as those used for multicore processors.

Different components of the integrated circuit, such as processor cores, non-core processing engines such as graphics engines, and other units are divided into different “domains,” and each domain is dynamically allocated a portion of an overall power budget for the integrated circuit.

Allocating an overall power budget for a processor in this way allows dynamic redistribution of power between the different domains, “enabling flexibility to handle various different workload requirements.”

Ex. 1001, 1:56 – 2:31; 10:25-28
POR, 4-5

'833 Patent

Allocation of the power budget between the separate power domains is handled by power management logic according to *sharing policy values* for each of the domains. The sharing policy values identify preferences for power allocation and are controllable by user-level software “to enable the software to determine how package power budget is shared between different domains.” Additionally, certain amounts of the power budget may be *reserved* for use by one or more domains.

The power management logic can account for not only the policy sharing values, but also the individual domains’ respective workload. The use of a central power management logic and policy sharing values allows a processor to shift power budget allocations between domains efficiently. For example, rather than components having set power allocations that are either used or unused, the power management logic can reallocate unused or inefficiently allocated power (e.g., in one domain) to other components (e.g., in a different domain).

Therefore, “embodiments provide a mechanism to dynamically share the power budget between different compute components in the same die. As 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, 1:55 – 2:31; 4:1-56; 5:2-15, 49-64; 6:29-39;
9:48-54; 10:16-18, 25-28
POR, 5-8

400

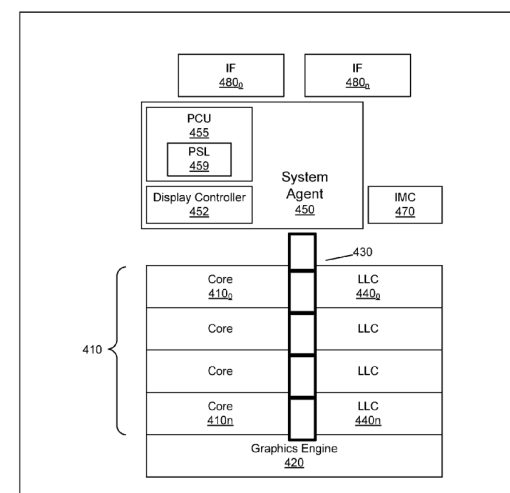


FIG. 7

'833 Claim Construction

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.

“independent” means “separate”

POR, 12-15
Ex. 2002, 32-35

'833 Challenges

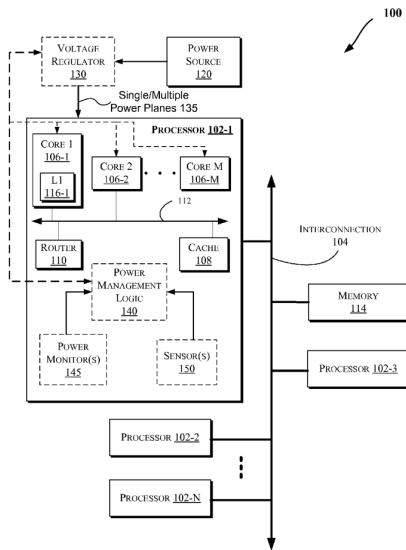
Claims Challenged	35 U.S.C. §	References
1-5, 7	103	Finkelstein
7	103	Finkelstein in view of Conroy
13-15, 17, 18	103	Nussbaum
16	103	Nussbaum in view of Bose
13-15, 17, 18	103	Conroy
16	103	Conroy in view of Bose
1-5,7, 13-15, 17, 18	103	Conroy in view of Finkelstein

Pet, 6-7

Finkelstein

(19) **United States**
 (12) **Patent Application Publication** (10) Pub. No.: **US 2010/0115304 A1**
 Finkelstein et al. (43) Pub. Date: **May 6, 2010**

(54) **POWER MANAGEMENT FOR MULTIPLE PROCESSOR CORES** Publication Classification
 (76) Inventors: **Lev Finkelstein, Hetanya (IL); Efraim Rotem, Haifa (IL); Aviad Cohen, Bat-Chen (IL); Ronny Ronen, Haifa (IL); Doron Rajwan, Rishon Le-Zion (IL)**
 (51) Int. Cl. **G06F 1/32 (2006.01)**
G06F 1/00 (2006.01)
 (52) U.S. Cl. **713/320; 327/513**
 (57) **ABSTRACT**
 Methods and apparatus relating to power management for multiple processor cores are described. In one embodiment, one or more techniques may be utilized locally (e.g., on a per core basis) to manage power consumption in a processor. In another embodiment, power may be distributed among different power planes of a processor based on energy-based considerations. Other embodiments are also disclosed and claimed.
 Correspondence Address: **Caven & Aghvili LLC, c/o CPA Global, P.O. BOX 52050, MINNEAPOLIS, MN 55402 (US)**
 (21) Appl. No.: **12/263,421**
 (22) Filed: **Oct. 31, 2008**



Finkelstein describes an approach to power distribution among different power planes under energy-based definitions.

[0028] In some embodiments, 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. Energy-based power management may be performed by controlling the energy budget defined iteratively as follows:

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

[0029] 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 , and α is the decay component. For example, using $\alpha=0.999$ corresponds to the window size of seconds, while using $\alpha=0.9$ corresponds to much smaller window size. The expression for E_n corresponds to the energy “remainder” which is the amount of energy not consumed by the system. The value of the decay component is defined by the requirements of the time window size. Also, the TDP values may be provided by a software application or user in some embodiments.

Ex. 1005 [0009], [0027], [0028]-[0029]

POR, 22-23

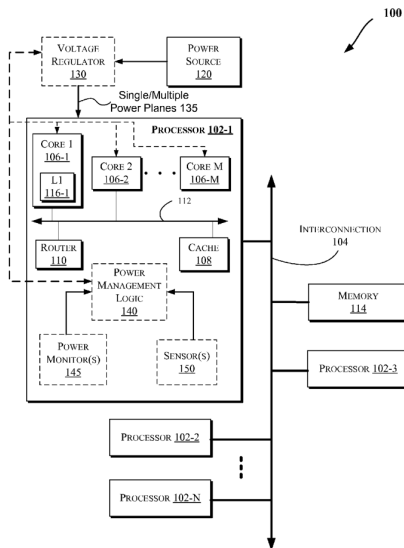
Finkelstein

(19) **United States**
 (12) **Patent Application Publication** (10) Pub. No.: US 2010/0115304 A1
 Finkelstein et al. (43) Pub. Date: May 6, 2010

(54) **POWER MANAGEMENT FOR MULTIPLE PROCESSOR CORES** Publication Classification
 (76) Inventors: Lev Finkelstein, Hetanya (IL); Efraim Rotem, Haifa (IL); Aviad Cohen, Bat-Chen (IL); Ronny Ronen, Haifa (IL); Doron Rajwan, Rishon Le-Zion (IL)
 (51) Int. Cl. G06F 1/32 (2006.01) G06F 1/00 (2006.01)
 (52) U.S. Cl. 713/320; 327/513

Correspondence Address: Caven & Aghevli LLC, c/o CPA Global, P.O. BOX 52050, MINNEAPOLIS, MN 55402 (US)
 (21) Appl. No.: 12/263,421
 (22) Filed: Oct. 31, 2008

ABSTRACT
 Methods and apparatus relating to power management for multiple processor cores are described. In one embodiment, one or more techniques may be utilized locally (e.g., on a per core basis) to manage power consumption in a processor. In another embodiment, power may be distributed among different power planes of a processor based on energy-based considerations. Other embodiments are also disclosed and claimed.



[0032] In an embodiment, a controller budget may be defined (e.g., implemented by the logic 140). Per power plane i , we define a controller function, denoted by $f^i(E)$, which maps energy budget onto the discrete set of power states. In one embodiment, a controller, which is required to be a non-decreasing function, maps a range $[E_{low}^i, E_{high}^i]$ onto the discrete range of $[P_n^i, P_0^i]$, where P_0^i is the maximal turbo state for this power plane, and P_n^i is the maximal efficiency state. Budget values below E_{low}^i are mapped into P_n^i , while budget values above E_{high}^i are mapped onto P_0^i . Other requirements may include meeting some requirements in anchor points—for example, the zero budget corresponds to the guaranteed power state called P_1 :

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

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

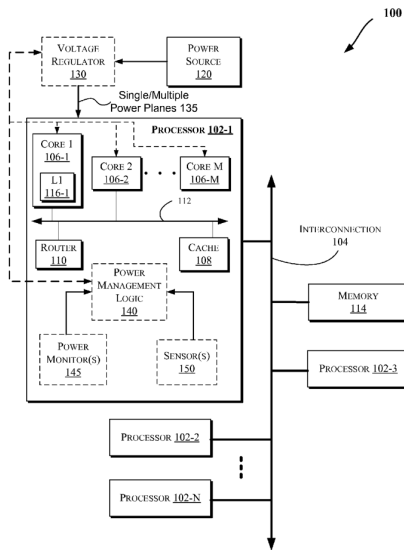
$$f^i(E) = P_0^i, E \geq E_{high}^i$$

Ex. 1005, [0031]-[0035]
 POR, 23-24

Finkelstein

(19) **United States**
 (12) **Patent Application Publication** (10) Pub. No.: US 2010/0115304 A1
 Finkelstein et al. (43) Pub. Date: May 6, 2010

(54) **POWER MANAGEMENT FOR MULTIPLE PROCESSOR CORES** Publication Classification
 (76) Inventors: Lev Finkelstein, Hetanya (IL); Efraim Rotem, Haifa (IL); Aviad Cohen, Bat-Chen (IL); Ronny Ronen, Haifa (IL); Doron Rajwan, Rishon Le-Zion (IL)
 (51) Int. Cl. G06F 1/32 (2006.01) G06F 1/00 (2006.01)
 (52) U.S. CL. 713/320; 327/513
 (57) **ABSTRACT**
 Methods and apparatus relating to power management for multiple processor cores are described. In one embodiment, one or more techniques may be utilized locally (e.g., on a per core basis) to manage power consumption in a processor. In another embodiment, power may be distributed among different power planes of a processor based on energy-based considerations. Other embodiments are also disclosed and claimed.
 Correspondence Address: Caven & Aghevli LLC, c/o CPA Global, P.O. BOX 52050, MINNEAPOLIS, MN 55402 (US)
 (21) Appl. No.: 12/263,421
 (22) Filed: Oct. 31, 2008



[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.

Failings of Finkelstein

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.

Finkelstein does not teach or suggest a dynamic allocation of a power budget *for a processor*, as claimed.

POR, 25

Failings of Finkelstein

Petition Argument

“Finkelstein discloses implementations in which an ‘unused’ budget for domain i , defined as a variable $W_{ki}E^k - E_{high}^i$, is ‘distributed among the rest of the power planes [domains] proportionally to their weights,’ defined as variables W_{ki} . A POSITA would have recognized such an allocation, where the power budget itself is a variable ($W_{ki}E^k - E_{high}^i$) and the variable power budget is being distributed based on variable weights (W_{ki}), as a ‘dynamic’ allocation because the allocation can change during run-time to correspond to the then-present operating conditions.”

Pet. at 24.

Flaws

Distributing ($W_{ki}E^k - E_{high}^i$), as alleged, is not allocating of an entire power budget for a processor, as required by claim 1.

POR, 25-26

Conroy

(19) United States
(12) Patent Application Publication (10) Pub. No.: US 2007/0049133 A1
Conroy et al. (43) Pub. Date: Mar. 1, 2007

(54) METHODS AND APPARATUSES FOR DYNAMIC POWER CONTROL

(51) Int. Cl. H01R 13/73 (2006.01)
(52) U.S. Cl. 439/894

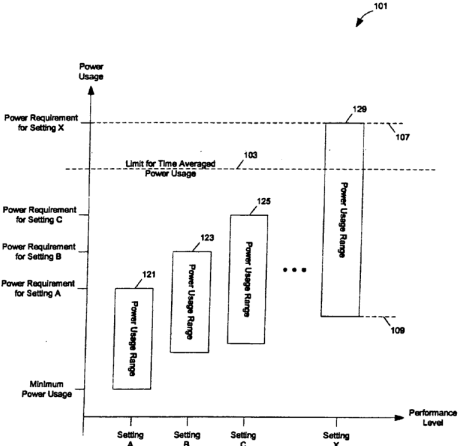
(76) Inventors: David G. Conroy, El Granada, CA (US); Michael Culbert, Monte Sereno, CA (US); Keith A. Cox, Campbell, CA (US)

Correspondence Address:
BLAKELY SOKOLOFF TAYLOR & ZAFMAN
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1030 (US)

(21) Appl. No.: 11/327,275
(22) Filed: Jan. 5, 2006

Related U.S. Application Data
(63) Continuation-in-part of application No. 11/212,970, filed on Aug. 25, 2005.

ABSTRACT
Exemplary embodiments of methods and apparatuses to dynamically redistribute power in a system that includes a plurality of subsystems are described. A load profile of the system is identified. The power is redistributed between the subsystems while tracking the load profile. The load profile may be an asymmetric, or a balanced load profile. The load profile is identified based on a utilization factor for each of the subsystems. In one embodiment, the power used by each of the subsystems is sensed by one or more sensors or predicted or estimated. A utilization factor, which may be a ratio of the actual power used by the subsystem to the power allocated to the subsystem, is calculated. The load profile is determined using the utilization factor of each of the subsystems. A power weighting arrangement between the subsystems, for example, a power distribution table, is selected based on the load profile.



[0006] In one embodiment, a load profile of the system is identified.

• • •

In one embodiment, the load profile is identified based on a utilization factor for each of the subsystems. In one embodiment, to determine the utilization factor, the actual power used by each of the subsystems is measured. The actual power used by each of the subsystems may be measured using one or more sensors coupled to the subsystems. Next, the utilization factor is calculated based on the measured power. In one embodiment, the utilization factor is calculated as a ratio of the actual power used by the subsystem to the power allocated to the subsystem. Alternatively, the load profile may be calculated as a ratio of the actual power used by the subsystem to the power allocated to the system.

Conroy

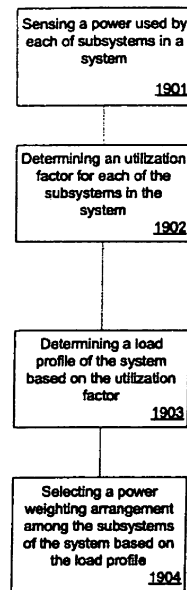


Fig. 19

1900

[0198] FIG. 19 is a flowchart 1900 of one embodiment of a method to dynamically redistribute power by tracking a load profile of a system. The method begins with operation 1901 by sensing an actual power used by each of subsystems in a system.

• • •

Next, in operation 1902, an utilization factor for each of the subsystems in the system is determined. The utilization factor may be considered as a power efficiency metric for a subsystem in the system.

• • •

[0199] Next, the method 1900 continues with operation 1903 which involves determining a load profile of the system based on the utilization factors of each of the subsystems.

• • •

[0200] After determining the load profile of the system, the method 1900 continues with operation 1904 which involves selecting a power weighting strategy (which may be considered a power weighting arrangement among the subsystems) of the system based on the load profile.

Conroy

[0201] In one embodiment, a power weighting strategy is selected based on the load profile by selecting a power distribution table out of a plurality of power distribution tables stored in a memory of the system.

Failings of Conroy

Petition Argument

Claim 13 is obvious in view of Conroy.

Conroy 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.” Examples of power distribution tables are illustrated in Figures 21A-C and 22, [] 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.

Pet. at 59-60.

Flaws

Claim 13 recites, in part:

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.

Conroy does ***not*** teach or suggest that the alleged “power sharing values” (i.e., the contents of the load profile tables—the power weighting arrangement settings) are stored in different storages.

In Conroy, all of the power distribution tables, and, therefore, all of the contents of each table, are stored in ***a memory***—that is, one memory. Conroy at [0201].

POR, 32-35

Failings of Conroy

Flaws

Petition Argument

“Conroy describes an example where [the] power settings are stored in registers.”

“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.”

Pet. at 61.

The “throttle settings” Conroy refers to are *not* the contents of the power distribution tables, but instead are the actual powers consumed by each subsystem.

Subsystems are instrumented, e.g., using sensors, to determine their actual power consumptions, e.g., by measuring currents and or voltages. The microcontroller stores the throttle settings determined by the sensor measurements in registers and uses that information to control the subsystems. Conroy at [0093]-[0097].

Rather than resort to using registers, a POSITA would simply adopt Conroy’s teaching of using a single storage to store the power distribution tables and their contents. Such an approach is in keeping with Conroy’s desire for simplicity and practices common in the art.

POR, 35-36

Failings of Conroy

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.

Petition Argument

Conroy uses “the past power usage (e.g., measurements ... at times $T-(n-1)\Delta$, $T-(n-2)\Delta$, ... , T)” to “dynamically determine the allowable power budget” for time interval $T+\Delta$, and allocates this “dynamically determined” power budget to different components of the system. A POSITA would have understood time interval $T+\Delta$ to be a “current time interval” because Conroy refers to the time intervals preceding $T+\Delta$ as being in the “past.”

Pet. at 63.

Flaws

A POSITA would appreciate that in the series, $T-(n-1)\Delta$, $T-(n-2)\Delta$, ... , T , the value “ T ” represents a *current time* and $T+\Delta$ is a *next time interval*.

[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 $T+\Delta$).

[0071] For example, the limit (203) may be a simple average in a time window (221) of a time period $(n+1)\Delta$ (e.g., less than 10 seconds).

POR, 37-39

Failings of Conroy-Bose

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.

Petition Argument

“[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.”

Pet. at 67.

Flaws

[0072] The system is then throttled to a setting that will not require power usage more than the dynamically determined budget (**205**).

[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.

In Conroy, subsystems do not *request* increases in system settings. Actual powers are used to make predictions about the need for future power. These predictions govern how power is allocated among the subsystems. Current settings are used until new power measurements reveal the need for redistribution of the allocated power.

POR, 39-41

Failings of Conroy-Bose

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.

Petition Argument

“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.”

Pet. at 67.

Flaws

Conroy conflicts with Bose:

While Conroy describes a system enacting a burst of high power following a period of low power, Bose describes the opposite—following low power usage, token count is decremented.

In Conroy, it is overall system performance that determines whether or not a change in operating parameters (as specified in a power distribution table) is employed, while in Bose it is component-level power considerations that determine token allocation.

POR, 41-42

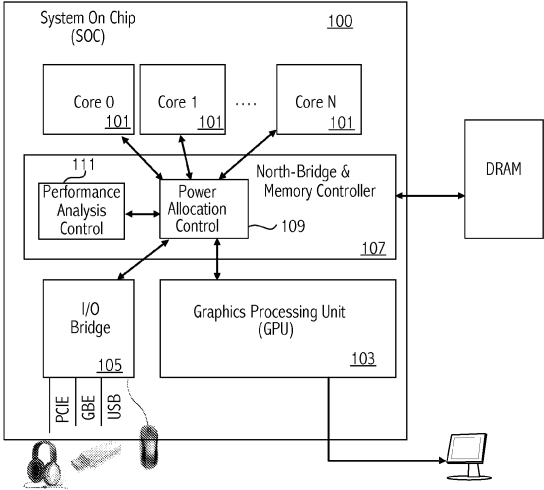
Nussbaum

(19)	United States	
(12)	Patent Application Publication	(10) Pub. No.: US 2011/0022356 A1
	Nussbaum et al.	(43) Pub. Date: Jan. 27, 2011
(54)	DETERMINING PERFORMANCE SENSITIVITIES OF COMPUTATIONAL UNITS	(52) U.S. CL. 702/182
(76)	Inventors: Sebastien Nussbaum , Lexington, MA (US); Alexander Branover , Chestnut Hill, MA (US); John Kalamatianos , Arlington, MA (US)	(57) ABSTRACT
	Correspondence Address: ZAGORIN O'BRIEN GRAHAM LLP (1001) 7600B NORTH CAPITAL OF TEXAS HIGHWAY, SUITE 350 AUSTIN, TX 78731-1191 (US)	Performance sensitivities to a change in performance capabilities of computational units of a computer system are determined based on measured utilization metrics for each of the computational units. In order to determine the performance sensitivities, in one approach, the computational units are operated at a first performance level, and respective first utilization metrics are determined. The computational units are then operated at a second performance level and respective second utilization metrics are determined. The sensitivity to performance capability change, e.g., a frequency change, is determined based on the respective first and second utilization metrics. The performance sensitivities of the computational units to a change in performance capability are continually updated in response to, e.g., a process context change of a computational unit or in response to a predetermined period of time elapsing since the last sensitivity to a performance capability change was determined for a computational unit.
(21)	Appl. No.: 12/508,902	
(22)	Filed: Jul. 24, 2009	
	Publication Classification	
(51)	Int. Cl. G06F 15/00 (2006.01)	

Nussbaum describes an “analysis of workload executed on computational units to help identify those computational units that are more performance sensitive to a change in performance capability caused by, e.g., a frequency change.”

“The computational units are operated at a first performance level, and respective first utilization metrics are determined. The computational units are then operated at a second performance level and respective second utilization metrics are determined. The sensitivity to performance capability change, e.g., a frequency change, is determined.”

Power distribution is then based on these measured frequency sensitivities.



Ex. 1007, [0005]-[0006], [0019].
POR, 43-44

Failings of Nussbaum

Petition Argument

Claim 13 is obvious in view of Nussbaum.

In Nussbaum, a power allocation controller and performance analysis control logic jointly control allocation of Thermal Design Point (TDP) power headroom.

The power headroom is selectively distributed among different computational units based on those computational units' sensitivities to frequency changes ("boost sensitivities"). The boost sensitivities govern how power is allocated between computational units and are "power sharing values" of domains, as recited in claim 13.

Pet. at 42, 44.

Flaws

In Nussbaum, boost sensitivities are not per-domain attributes. Instead, they are per-component, e.g., per-core, attributes:

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.

Nussbaum at [0025].

Nussbaum criticizes domain-based power allocations: "treat[ing] all active units homogeneously when dithering frequency or boosting steady state frequency" is undesirable because "some active cores or other computational units may be gaining little or no performance increase." Nussbaum at [0019].

Failings of Nussbaum

Claim 17 depends from claim 14 and recites, “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.”

Petition Argument

Claim 17 is obvious in view of Nussbaum.

The recited “minimum reservation values” are met by Nussbaum’s disclosure of a “core power state,” referred to as “Deep Cstate,” in which a core is allocated “Core_DeepCState_Pwr.”

Pet. at 49.

Flaws

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

Horst deposition:

1	A. Yes, effectively, the processor isn't executing	
2	anything in that state.	

A state in which a processor is off does not correspond to a state where the processor has been allocated a minimum reservation value of power, which the specification of the '833 patent explains is a reserved amount of power to be allocated a domain.

POR, 50-51
Ex. 2001, 44:1-2

Failings of Nussbaum

Claim 17 depends from claim 14 and recites, “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.”

Petition Argument

Claim 17 is obvious in view of Nussbaum.

“[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.”

“[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.”

Pet. at 49-50.

Flaws

In Nussbaum, to the extent power is allocated to cores or GPUs in accordance with performance states, those allocations are made on a per-computational unit basis, not a per-domain basis.

POR, 51-52

Failings of Nussbaum

Claim 18 depends from claim 14 and recites, “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.”

Petition Argument

Claim 18 is obvious in view of Nussbaum.

“[I]f 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.”

“[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).”

Pet. at 50-51.

Flaws

In Nussbaum, 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.

The output value is used to throttle CPU cores and create more power budget, but no power budget is allocated using the output of step 601. The output of step 601 is simply an identification.

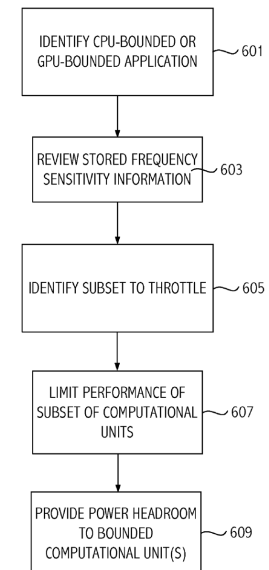


FIG. 6

Failings of Nussbaum-Bose

Claim 16 depends from claim 14 and recites, “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.”

Petition Argument

Claim 16 is obvious in view of Nussbaum.

“A POSITA would have been motivated to implement Nussbaum’s power allocation operations with Bose’s teaching to increment a boost sensitivity of a particular domain ... 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.”

Pet. at 54.

Flaws

Nussbaum and Bose would not have been combined. Nussbaum is rooted in physical solutions and uses “boost sensitivities” that depend on physical characteristics of computational units, whereas Bose is rooted in software solutions and uses arbitrary “tokens” of power allocation.

And, the combination does not teach or suggest “increment[ing] the first power sharing value when a request for a higher frequency for the first domain is not granted.” Nussbaum’s “boost values” are calculated based on the performance level and utilization metrics. They are not responsive to requests. In Bose, when no additional tokens can be allocated to a particular resource when requested, that resource must wait. So, if anything, the combination teaches waiting for an empirical reevaluation before altering a boot value.

Failings of Conroy-Finkelstein

Petition Argument

Claims 1-5, 7, 13-15, and 17-18 are obvious in view of Conroy when considered in combination with Finkelstein.

“[A] POSITA would have been motivated to implement Conroy’s system on a chip, as in Finkelstein.”

Pet. at 68 *et seq.*

Flaws

Modifying Conroy with Finkelstein would merely add another (Finkelstein-like) component within Conroy’s system but would not change the overall power management characteristics of Conroy.

Conroy is agnostic when it comes to the manner in which power is allocated at the level of individual components/subsystems.

A POSITA would therefore appreciate that there is no need to modify the power allocation approach of Conroy if and when components were added. Hence, power would be redistributed between sub-systems while tracking a load profile.

Failings of Conroy-Finkelstein

Petition Argument

Claim 1 is obvious in view of Conroy-Finkelstein.

In Conroy “subsystem A” and “subsystem B” serve as the claimed first and second domains.

Pet. at 71.

Flaws

There is no suggestion in Conroy that subsystems A and B operate at *independent* voltage and frequency.

While “components have independent throttle settings,” Conroy at [0120], those throttle settings are merely variable, not separately controlled.

Conclusion

The Petition has failed to prove the unpatentability of any of the challenged claims.

POR, 61