UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MEDIATEK INC.,
Petitioner,

v.

DAEDALUS PRIME LLC,
Patent Owner

Case No. IPR2025-00243
U.S. Patent No. 8,769,316

PETITION FOR *INTER PARTES* REVIEW

TABLE OF CONTENTS

Page(s)

TABLE OF AUTHORITIES

Page(s)

Federal Cases

Federal Statutes

Regulations

LIST OF EXHIBITS

Exhibit	Description
1001	U.S. Patent No. 8,769,316 to Ananthakrishnan et al. ("'316 patent")
1002	Prosecution History of the '316 patent
1003	Declaration of Murali Annavaram. PhD ("Annavaram")
1004	Curriculum Vitae of Annavaram
1005	U.S. Patent Appl. Pub. No. 2010/0115304 to Finkelstein et al. ("Finkelstein")
1006	U.S. Patent Appl. Pub. No. 2006/0288241 to Felter et al. ("Felter")
1007	U.S. Patent Appl. Pub. No. 2010/0162006 to Therien et al. ("Therien")
1008	Linley Gwennap, *Sandy Bridge Spans Generations: Intel Focuses on Graphics, Multimedia in New Processor Design*, Microprocessor Report, September 2010 ("Gwennap")
1009	U.S. Patent Appl. Pub. No. 2009/0328055 to Bose et al. ("Bose")
1010	U.S. Patent Appl. Pub. No. 2008/0288748 to Sutardja et al. ("Sutardja")
1011	U.S. Patent Appl. Pub. No. 2007/0049133 to Conroy et al. ("Conroy")
1012	U.S. Patent Appl. Pub. No. 2009/0322768 to Lalonde et al. ("Lalonde")
1013	Complaint, *Daedalus Prime LLC v. MediaTek Inc.*, No. 2:24-cv-00235-JRG, Dkt. 1 (E.D. Tex. Apr. 8, 2024)
1014	Docket Control Order, *Daedalus Prime LLC v. MediaTek Inc.*, No. 2:24-cv-00235-JRG, Dkt. 33 (E.D. Tex. Oct. 23, 2024)
1015	Plaintiff's Infringement Contentions, *Daedalus Prime LLC v. MediaTek Inc.*, No. 2:24-cv-00235-JRG

LIST OF CHALLENGED CLAIMS

Claim 8	
[8Pre]	A method comprising:
[8a]	determining, in a power controller of a multi-domain processor, a power budget for the multi-domain processor for a current time interval, the multi-domain processor including at least a first domain and a second domain;
[8b]	determining, in the power controller, a portion of the power budget to be allocated to the first and second domains, including
[8c]	allocating a minimum reservation value to the first domain and a minimum reservation value to the second domain, and
[8d]	sharing a remaining portion of the power budget according to a first sharing policy value for the first domain and a second sharing policy value for the second domain; and
[8e]	controlling a frequency of the first domain and a frequency of the second domain based on the allocated portions.
Claim 12	
The method claim 8, wherein the first sharing policy value is to be incremented when a request for a higher frequency for the first domain is not granted, and the second sharing policy value is to be incremented when a request for a higher frequency for the second domain is not granted.	

Claim 13	
[13Pre]	A system comprising
[13a]	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,
[13b]	the third domain to operate at a fixed power budget and to dynamically allocate a variable power budget between the first and second domains, wherein
[13c]	the system agent circuitry includes a power sharing logic to determine the variable power budget for a current time interval and
[13d]	to allocate a first portion of the variable power budget to the first domain according to a first power sharing value for the first domain, and to allocate a second portion of the variable power budget to the second domain according to a second power sharing value for the second domain; and
[13e]	a dynamic random access memory (DRAM) coupled to the multicore processor.
Claim 14	
The system of claim 13, 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.	
Claim 15	
The system of claim 13, 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.	

Claim 16
The system of claim 13, 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.
Claim 17
17. The system of claim 13, 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.

I. Introduction

Petitioner MediaTek Inc. requests *inter partes* review of claims 8 and 12-17 of U.S. Patent No. 8,769,316 (Ex-1001, "'316 patent"), assigned to Patent Owner Daedalus Prime LLC, and cancellation of the same based on the following:

Grounds	Claims	Basis	Prior Art
1	8, 12	§103	Felter
2	13, 14, 17	§103	Finkelstein
3	8, 12-17	§103	Finkelstein, Felter
4	12, 15, 17	§103	Finkelstein, Felter, Therien
5	12	§103	Felter, Therien
6	15, 17	§103	Finkelstein, Therien

II. '316 Patent

The '316 patent was filed on September 6, 2011, and issued on July 1, 2014. It discloses "a power balancing mechanism" for allocating "a package power budget … between the different domains of a multicore processor."[1] *Id.*, 2:7-16. Figure 1 (below) illustrates "a high level method of performing power budget allocations between multiple domains." *Id.*, 5:11-14.

[1] Domains are "also referred to as 'planes'." Ex-1001, 4:2-5.

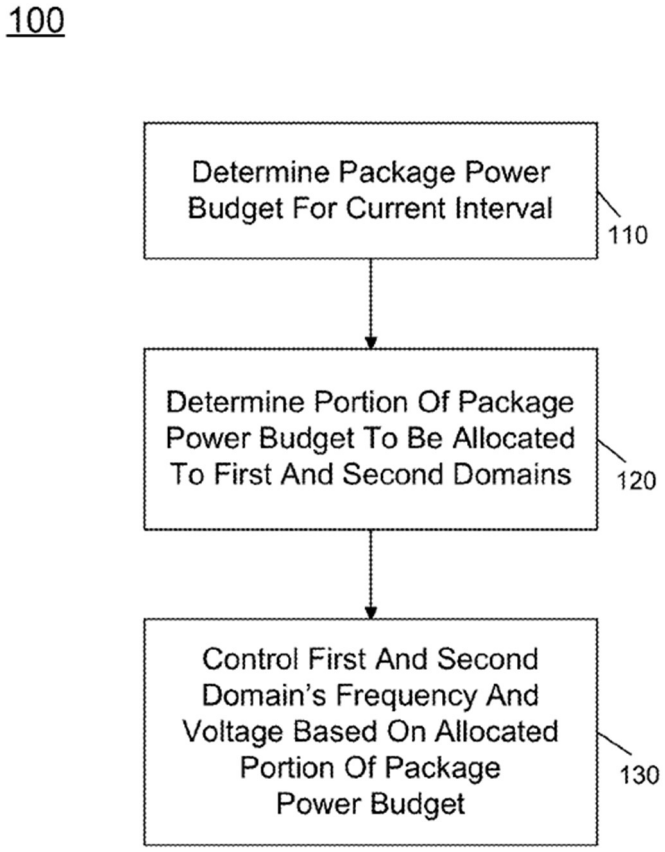


FIG. 1

Id., FIG. 1.

As Figure 7 illustrates below, the multicore processor "can include at least two independent domains, one associated with the cores … and one associated with a graphics engine." *Id.*, 1:59-62; *see id.*, 8:21-26, FIG. 7. It may further include a "system agent domain … to handle power control events and power management." *Id.*, 9:5-17.

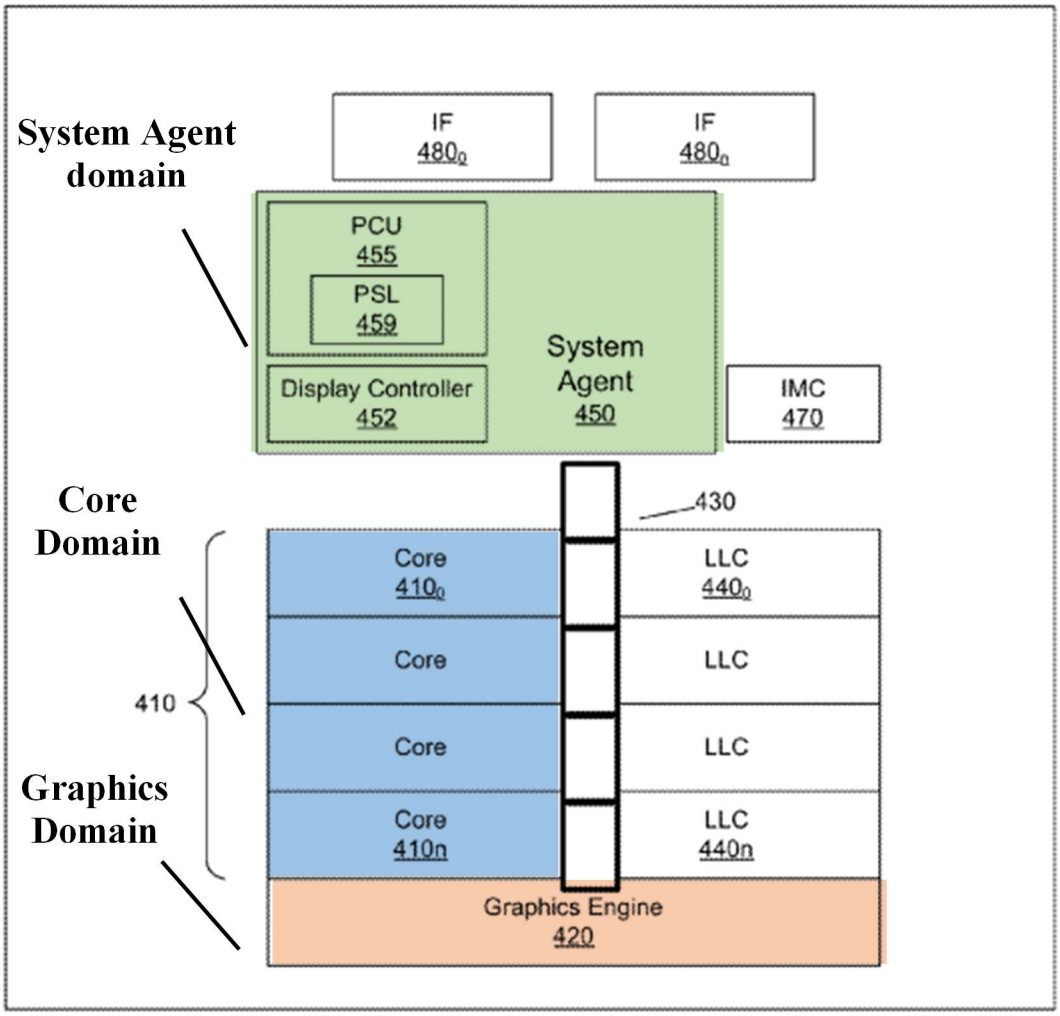


FIG. 7

Id., FIG. 7 (annotated).

"[A] power budget management (PBM) algorithm may be executed by logic such as logic of a power control unit (PCU) of a processor" based on "various

processor parameters." *Id.*, 2:28-33; *see* 3:63-4:51, 5:3-10, 7:1-49. For example, the algorithm may "first allocate to the graphics domain and then to the core domain [and] share the rest between core domain and graphics domain." *Id.*, 6:26-29; *see id.*, 8:47-52.

III. Level of Ordinary Skill in the Art

A person of ordinary skill in the art ("POSITA") would have held at least a bachelor's degree in electrical engineering, computer engineering, physics, or similar field and approximately two years of industry or academic experience related to microelectronic architecture design and development, including experience with power management in multicore processors. Work experience could substitute for formal education, and vice versa. Annavaram ¶54.

IV. Claim Construction

No terms need construction to resolve the issues in this Petition.[2] *See Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017).

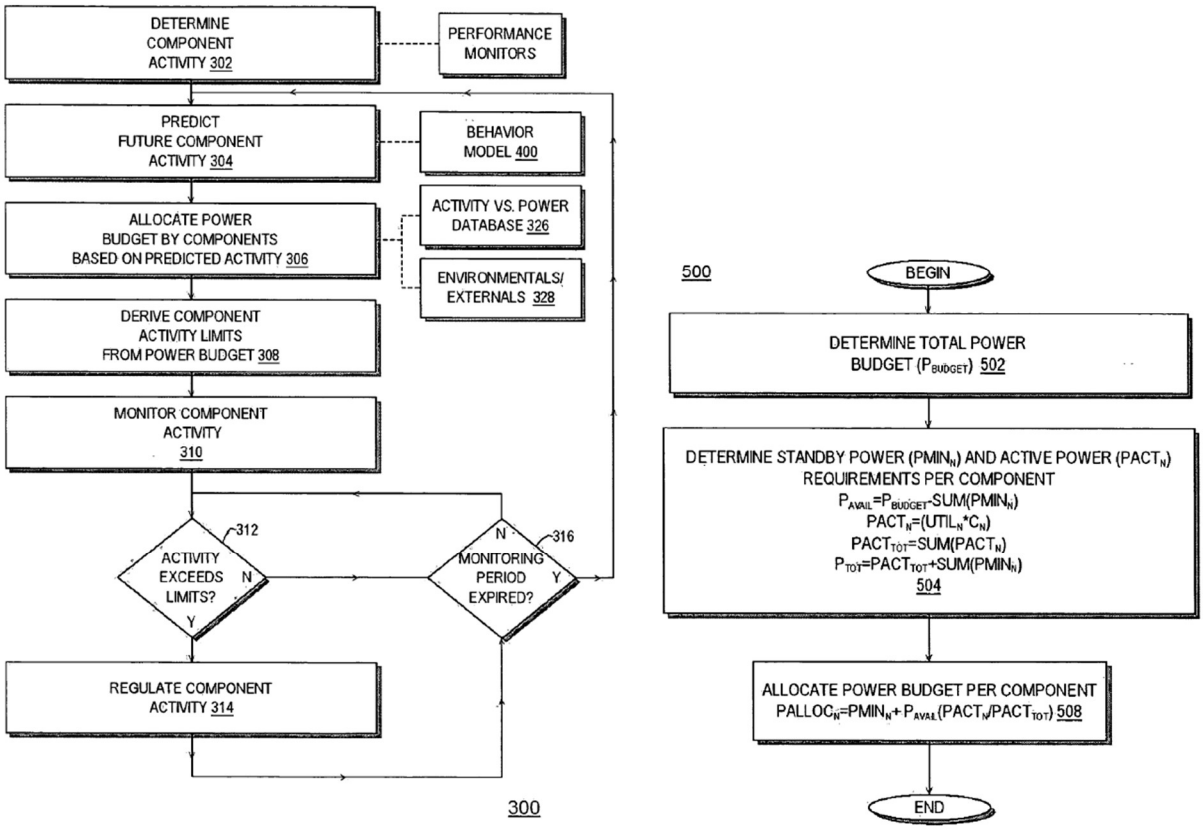
V. Ground 1: Felter Renders Obvious Claims 8 and 12

A. Overview of Felter (Ex-1006)

Felter, published on December 21, 2006, is prior art under §102(b). Like the '316 patent, Felter discloses a "method of dynamically managing power in a multi

[2] Claim construction may be needed to resolve different disputes elsewhere.

component system" (Figure 3, below) and "allocating power budgets to the system components based on expected levels of activity" (Figure 5, below). Felter, [0014], [0016]; *see id.*, Abstract, [0057]-[0066], FIGS. 3, 5.



Felter, FIGS. 3, 5.

Felter implements those techniques using a "power administrator" in a "single-chip system" with at least two processors.[3] Felter, [0012], [0026]-[0028], [0035]-[0049], FIGS. 1-2.

[3] Processor 102 in Figure 2 below represents two processors 102-1 and 102-2. *See* Felter, [0026], FIG. 1.

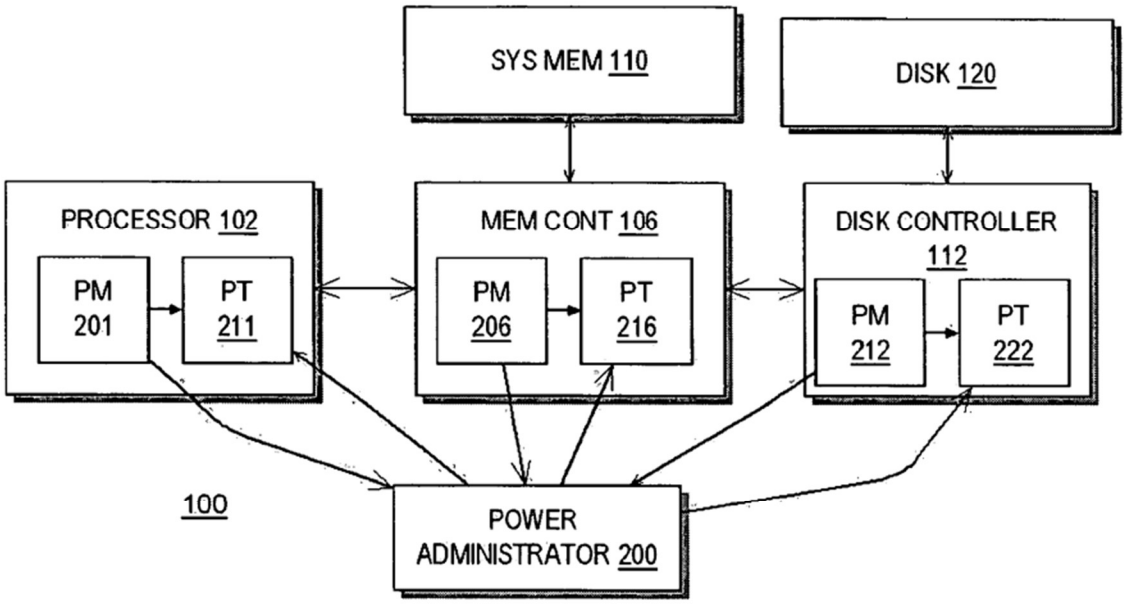


FIG. 2

Felter, FIG. 2.

The power administrator generates a "power estimate for each component" and "then allocate[s] the budgeted power amongst all the components." Felter, [0019]. This "includes allocating each component its corresponding standby power and a share of the system power available for dynamic powering based on the expected levels of activity." Felter, Abstract, [0010]; *see* [0057]-[0061]. The power administrator also "monitors component activity using performance monitors (PM's)" and "throttles" activity as needed with "performance throttles (PT's)." Felter, [0030]-[0031]; *see* [0059], [0065]-[0066].

Felter is analogous art because it is in the same field of endeavor as the '316 patent (power allocation for microprocessors) and reasonably pertinent to a

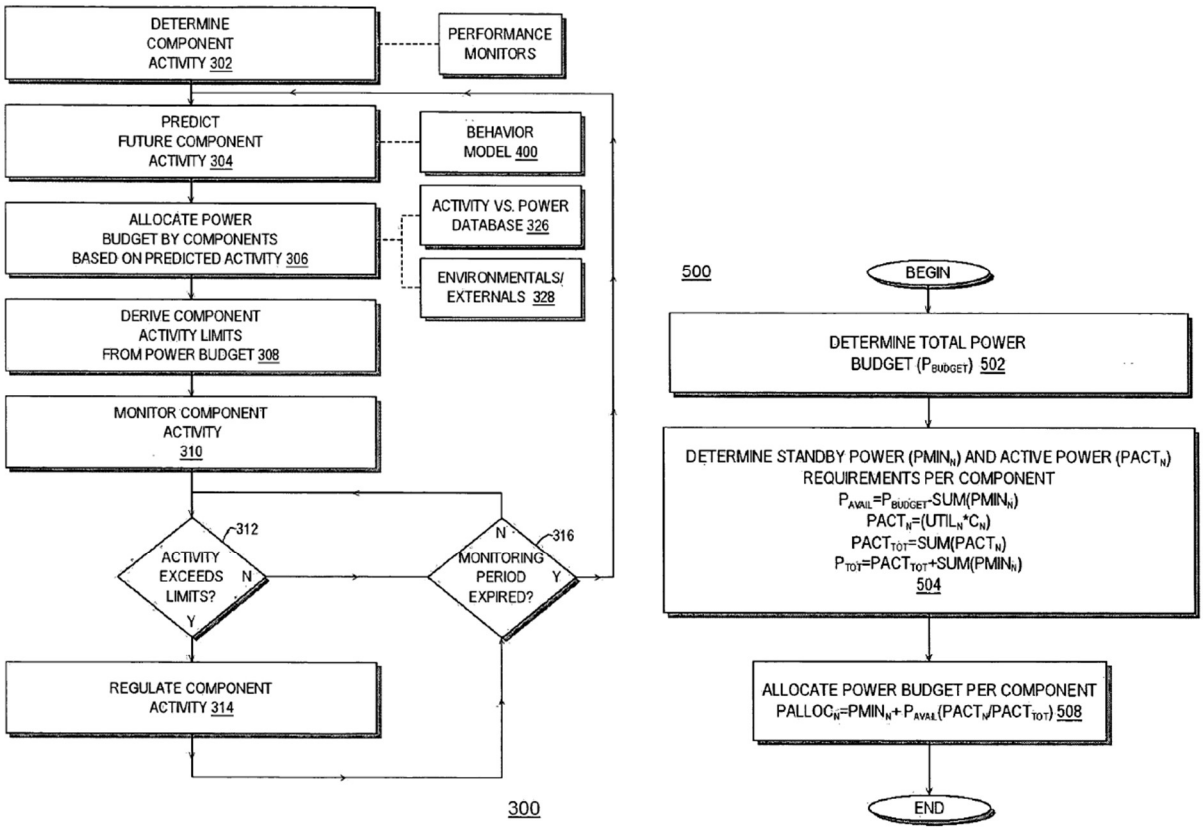
problem it addresses (ensuring sufficient power). *Compare* Felter, [0001]-[0002], [0007], [0009], *with* Ex-1001, Abstract, 1:18-22; *see* Annavaram ¶66.

B. Claims

1. Independent claim 8

a. [8pre]: "A method comprising:"

Felter discloses [8pre] because it discloses a "method for managing power in a data processing system" by "allocating power budgets to the system components based on expected levels of activity." Felter, Abstract, [0016]; *see* [0057]-[0061], FIGS. 3, 5; Annavaram ¶73.



Felter, FIGS. 3, 5.

b. [8a]: "determining, in a power controller of a multi-domain processor, a power budget for the multi-domain processor for a current time interval, the multi-domain processor including at least a first domain and a second domain"

Felter discloses [8a] because it discloses a "single-chip system[]" with multiple "processor cores" operating on independent power domains (multi-domain processor) and a "power administrator" (power controller) to allocate, for each upcoming time interval, a "power budget (P_{BUDGET})" among the system components. Felter, [0028], [0030], [0057]; *see* [0026], [0035]-[0049], [0057]-[0066], FIGS. 1-5; Annavaram ¶74.

The '316 patent uses "'domain' … to mean a collection of hardware and/or logic that operates at the same voltage and frequency point." Ex-1001, 1:52-56; Annavaram ¶75. Felter's Figure 1 below highlights two processors 102-1 and 102-2. Felter, [0026]. They correspond to first and second domains, respectively, because they independently operate at distinct operating voltages and frequencies. *See* Felter, [0034] ("[P]rocessors 102 may support voltage and frequency scaling …."), [0042] (calling them "two independent frequency-scalable processors"); Annavaram ¶74.

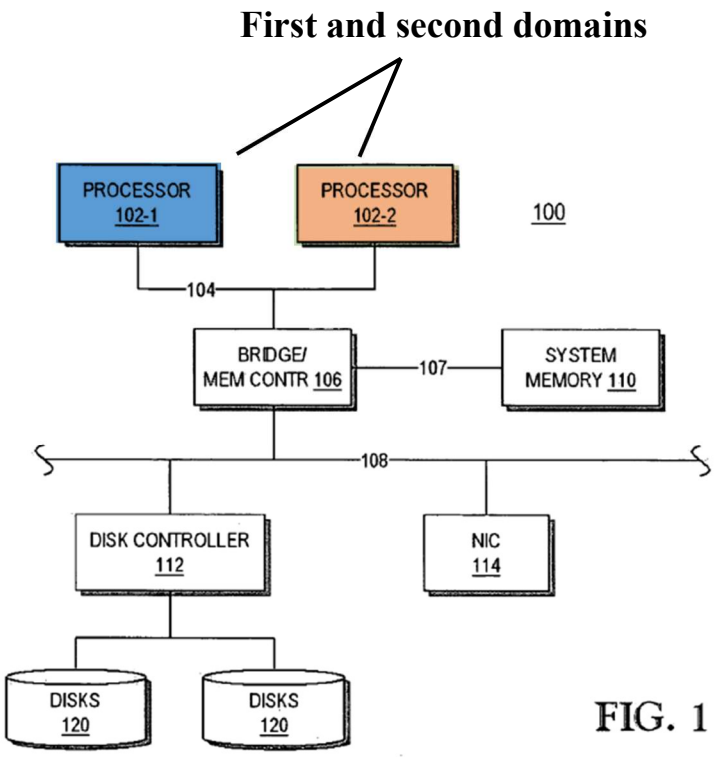
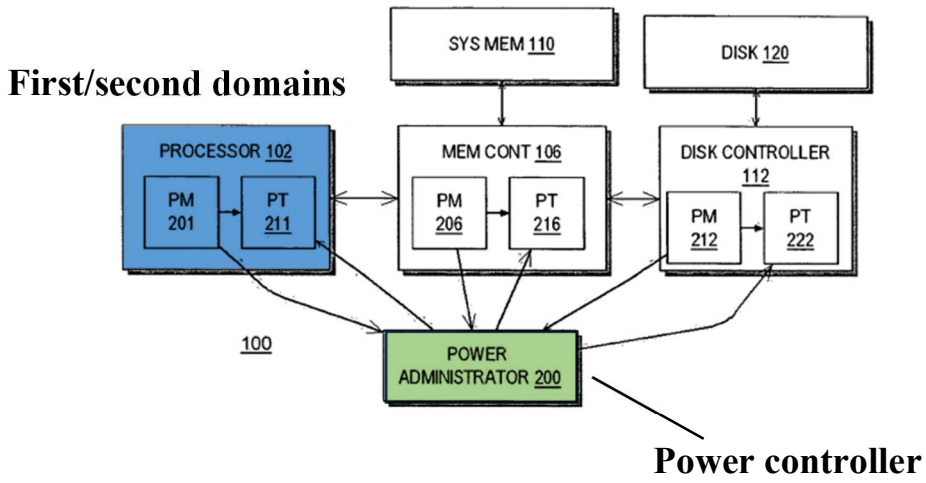


FIG. 1

Felter, FIG. 1 (annotated).

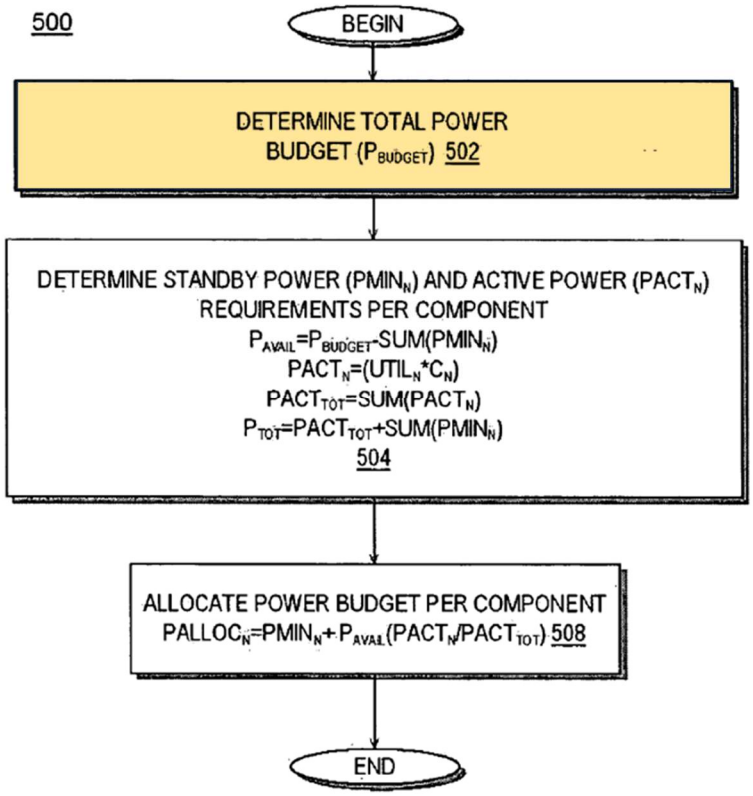
Felter's "dynamic power allocation administrator" determines a "power budget (P_{BUDGET}) of the system" for "the upcoming [time] interval." Felter, [0057], [0059]; *see id.*, [0057]-[0061], FIGS. 2, 5. It then ensures "power is allocated to the various components us[ing] a modified pro rata technique." Felter, [0061]; *see id.*, FIGS. 2, 5; Annavaram ¶76. "When the current interval expires, power administrator 200 generates new predictions for the next upcoming period … and the process repeats itself indefinitely." Felter, [0066].

Annotated Figure 2 below highlights the first and second domains and the "power administrator" (power controller). Felter, FIG. 2, [0029] (referencing "the processors 102 of FIG. 1"); Annavaram ¶76.



Felter, FIG. 2 (annotated).

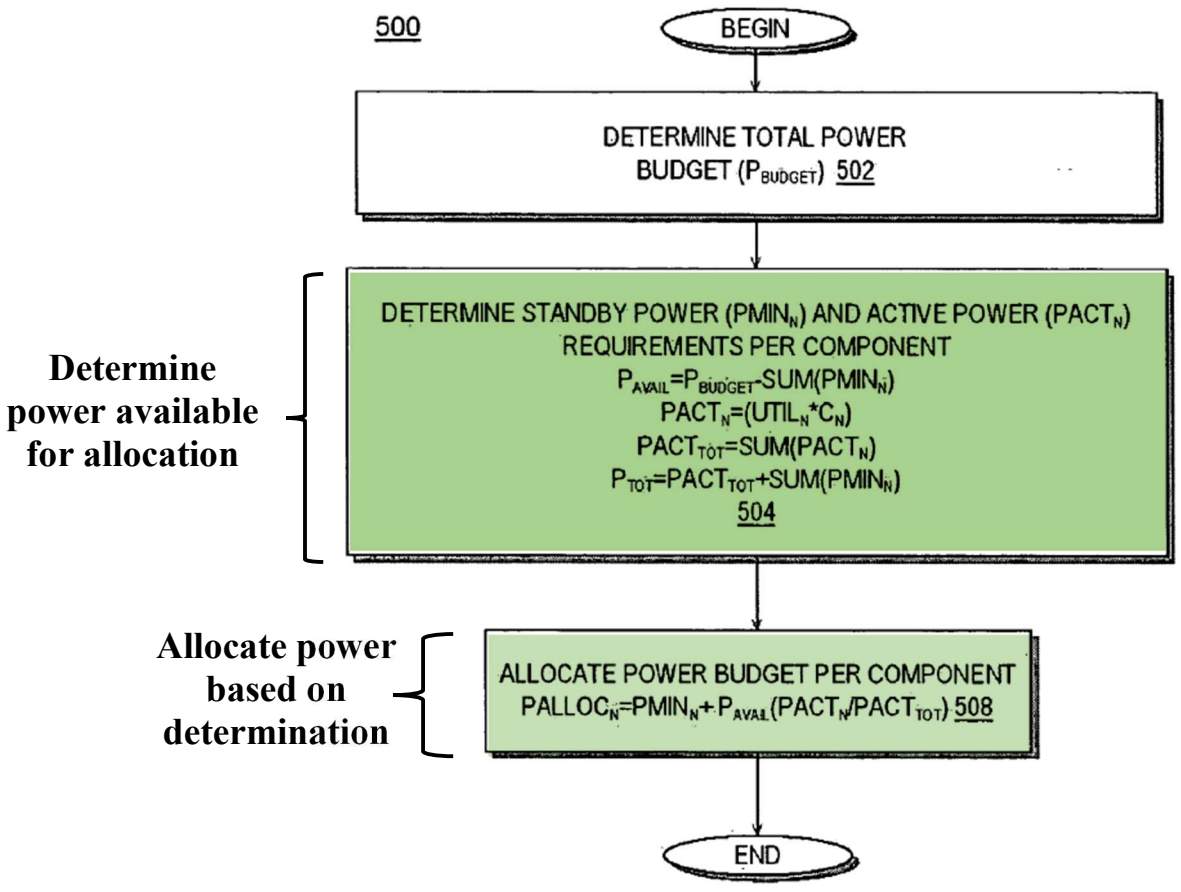
Felter's Figure 5 below illustrates an exemplary process for budgeting and allocating power for the current time interval. It begins by determining the power budget (P_{BUDGET}). *See* Felter, [0057], FIG. 5; Annavaram ¶77.



Felter, FIG. 5 (annotated).

c. [8b]: "determining, in the power controller, a portion of the power budget to be allocated to the first and second domains, including"

Felter discloses [8b] because its power administrator (power controller) ensures "power is allocated to the various components us[ing] a modified pro rata technique." Felter, [0061]; *see id.*, [0019], [0030], [0057]-[0066], FIG. 5; Annavaram ¶¶78-80. This includes allocating power to the "processor cores" operating on the first and second domains. Felter, [0028]; *see* [0026], [0042]-[0049], [0061], FIG. 5.



Felter, FIG. 5 (annotated).

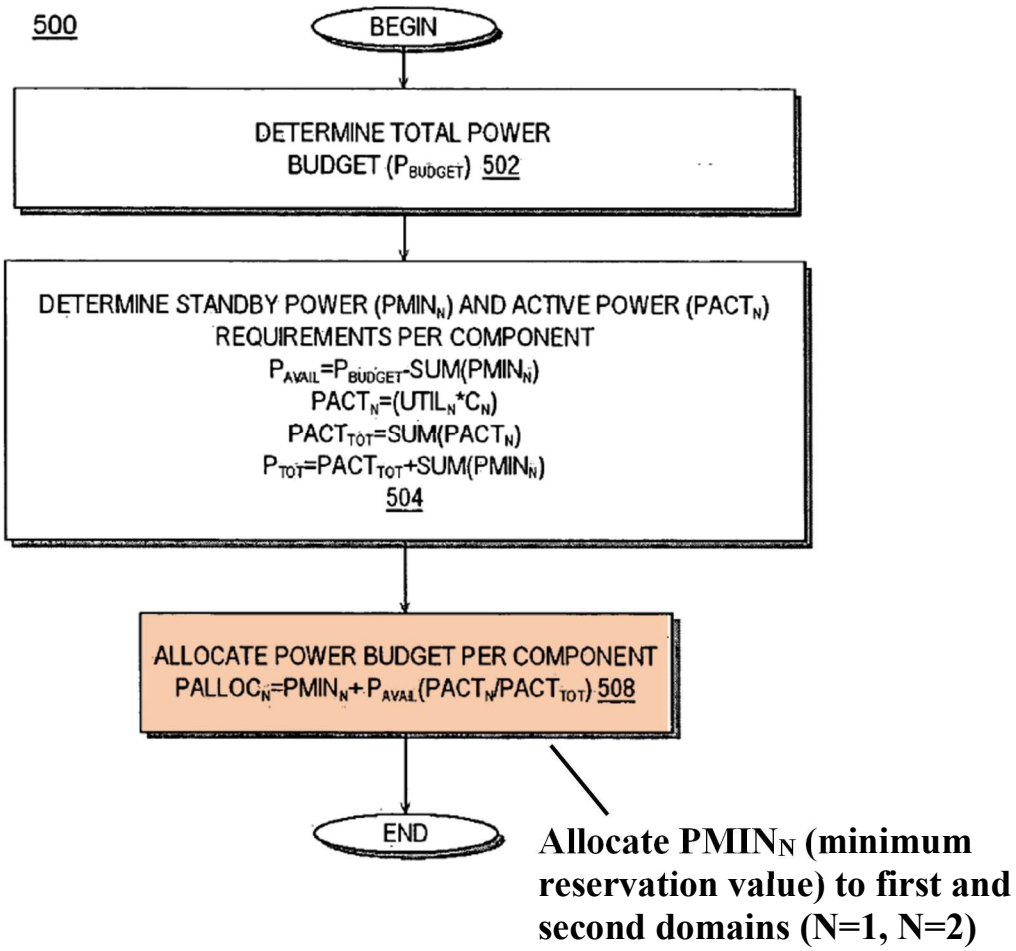
As Figure 5 above illustrates, "the amount of power budgeted for each component is equal to the component's standby power requirements (which must always be met) and a portion of the available power P_{AVAIL}." Felter, [0061]. The Nth domain is allocated a portion ($PALLOC_N$) of the power budget (P_{BUDGET}). Felter, [0061], FIG. 5. This allocation "distributes the available system power to the system components in a manner that (1) ensures that each component receives a minimum level of power and (2) allocates the available active power in proportion to the requirement for the components by the workload(s) executing on the system." Felter, [0061]; *see* Annavaram ¶¶79-80.

d. [8c]: "allocating a minimum reservation value to the first domain and a minimum reservation value to the second domain, and"

Felter discloses [8c] because its power administrator allocates each component its "standby power requirements (which must always be met)." Felter, [0061]; *see* [0039], [0042]-[0049], [0065], FIG. 5; Annavaram ¶¶81-83. The standby power requirement for the Nth domain (its minimum reservation value) is called "$PMIN_N$." Felter, [0058], FIG. 5.

Each component (including each processor core) has a "fixed" $PMIN_N$ corresponding to "the minimum operational power of the processor." Felter, [0039], [0042]-[0049]. After the power administrator determines "the power budget and the active and standby demands of each of the components" (*see* Felter, [0058]-[0060],

FIG. 5 (block 504)), it allocates to each component its "standby power requirements (which must always be met) and a portion of the available power P_{AVAIL}." Felter, [0061], FIG. 5.



Felter, FIG. 5 (annotated).

In this way, the power administrator "ensure[s] there is always enough power for each component to obtain some minimal work from it" by including "a reservoir for minimal component activity." Felter, [0065]. Thus, Felter discloses allocating a minimum reservation value ($PMIN_N$) to each of the processor cores (first and second domains, N=1 and N=2). Annavaram ¶¶82-83.

e. [8d]: "sharing a remaining portion of the power budget according to a first sharing policy value for the first domain and a second sharing policy value for the second domain; and"

Felter discloses [8d] because its power administrator (power controller) allocates power to the processor cores using a "modified pro rata technique," whereby "the amount of power budgeted for ***each component*** is equal to ***the component's standby power requirements (which must always be met) and a portion of the available power*** P_{AVAIL}." Felter, [0057]-[0061] (emphasis added); Annavaram ¶¶84-88.

After determining a minimum reservation value ($PMIN_N$) for each domain, Felter determines the remainder of the power budget (P_{AVAIL}): "[A]n available power figure (P_{AVAIL}) is determined by subtracting from the maximum power deliverable by the system the sum of all the standby values." Felter, [0058]; *see id.*, FIG. 5 ("$P_{AVAIL}=P_{BUDGET} - SUM(PMIN_N)$"). P_{AVAIL} thus represents a "remaining portion of the power budget" as claimed. Annavaram ¶85.

Each domain is allocated a portion of P_{AVAIL}: "The portion of P_{AVAIL} allocated to a system is determined by the ratio of the components active power requirements to the total active power requirements of the system. This formula is represented in block 508 as $PALLOC_N=PMIN_N+P_{AVAIL}*(PACT_N/PACT_{TOT})$." Felter, [0061]; *see* FIG. 5. The values $PACT_N/PACT_{TOT}$ represent the claimed "sharing policy value[s]." Annavaram ¶86. By allocating the remaining power according to these

sharing policy values, the power administrator "allocates the available active power [P_{AVAIL}] in proportion to the requirement for the components by the workload(s) executing on the system." Felter, [0061]; *see* [0050]; Annavaram ¶87.

The total power allocated to each domain N is:

$$\mathbf{PALLOC_N=PMIN_N+P_{AVAIL}*(PACT_N/PACT_{TOT})},$$

where $PACT_N/PACT_{TOT}$ represents the first and second sharing policy values for the first and second domains (N=1, N=2). Felter, [0061], FIG. 5; *see* Annavaram ¶88. $PACT_N$ is "the product of the activity (UTIL) predicted for component N ($UTIL_N$) and a constant of proportionality for component N (C_N)." Felter, [0058]-[0059], FIG. 5 (block 504). Because $PACT_{TOT}$ is the sum of all $PACT_N$, the sharing policy values $PACT_N/PACT_{TOT}$ ensure each domain N shares power proportional to its expected utilization. Felter, [0059]-[0061], FIG. 5; Annavaram ¶88.

f. [8e]: "controlling a frequency of the first domain and a frequency of the second domain based on the allocated portions."

Felter discloses [8e] because it discloses regulating component (domain) activity using "voltage-frequency scaling." Felter, [0036]-[0049], [0065]; Annavaram ¶¶89-91. The power administrator "enforces the allocated power budget by monitoring the activity of the various components and constraining or throttling the activity of any component that exceeds its power allocation." Felter, [0030]; *see* [0031], [0033], [0057], FIGS. 2, 5. Thus, when a component, including the first or

second processor, "is subject to an activity limit [and] reaches that limit, further activity is prohibited until the interval is over." Felter, [0033]. To do so, Felter discloses using "dynamic voltage and frequency scaling (DVFS)" (also called voltage-frequency scaling / throttling), a conventional way of controlling frequency. Felter, [0035]-[0036]; *see* [0037]-[0049], [0065]-[0066]; Annavaram ¶89-91.

Accordingly, Felter discloses controlling the frequencies of the first and second domains based on the allocated portions of P_{BUDGET}.

2. Claim 12: "The method claim 8, wherein the first sharing policy value is to be incremented when a request for a higher frequency for the first domain is not granted, and the second sharing policy value is to be incremented when a request for a higher frequency for the second domain is not granted."

Felter discloses claim 12 because it discloses "throttling" (not granting) a request to perform processor core activity that would "exceed" its activity "limit" during a first interval. Felter, [0065]. After denying attempts to perform more instructions during that interval (denying a request to perform at a higher frequency), a POSITA would have at least found it obvious to increase a processor core's frequency during a subsequent interval by allocating more power to it. *See* Felter, [0051] (explaining "that very recent activity is a good predictor of current activity"), [0066] ("When the current interval expires, power administrator 200 generates new predictions for the next upcoming period … and the process repeats itself indefinitely."); Annavaram ¶¶92-97.

As explained for [8c], Felter allocates power to first and second processor components (domains) by accounting for "the component's standby power requirements (which must always be met) and a portion of the available power P_{AVAIL}." Felter, [0057]-[0061]. Felter's power administrator then "deriv[es] (block 308) component activity limits that correspond to the power budgets allocated in block 306." Felter, [0064], FIG. 3; *see* Annavaram ¶¶93-94. If any components exceed their limit, the power administrator throttles activity "until the next monitoring interval begins." Felter, [0064]-[0066]; Annavaram ¶94.

Felter explains "that very recent activity is a good predictor of current activity." Felter, [0051]. Thus, in one example, "the predicted activity for time period N+1 is a simple function of the monitored activity for time period N." *Id.* This approach improves "responsiveness in the context of a rapidly changing operating environment." *Id.* Using recent activity to predict current utilization, a POSITA would have understood that if one of Felter's power domains was throttled during the previous time interval because it lacked sufficient power, then its predicted utilization ($UTIL_N$) should be incremented for the next interval to permit increased activity. *See* Annavaram ¶95; Felter, [0051], [0054] ("[P]ower administrator 200 includes allocating (block 306) power budgets to the various system components based on the predicted level of activity."), [0066] ("When the current interval expires, power administrator 200 generates new predictions for the

next upcoming period in block 304 and the process repeats itself indefinitely."), FIG. 3. That is, if one of Felter's components (domains) was prevented from performing more operations during the previous time interval (denying its request to perform at a higher frequency), then a POSITA would have understood it should be granted a higher frequency during the next interval. *See* Annavaram ¶95; Felter, [0035]-[0036] ("voltage-frequency scaling"), [0065]-[0066] ("voltage-frequency throttling").

Incrementing was a known, suitable option to increase frequency. As Dr. Annavaram explains, it was well known to increase the power budget of a domain when its power needs are not satisfied. Annavaram ¶96 (citing Therien, [0031] (describing a "software policy manager" that "periodically evaluates" a system "to determine whether any power budget adjustments are appropriate" and "make[s] budget adjustments as conditions associated with the system 400 change (e.g., when a component raises or lowers an increase request flag)"); Conroy, [0006] ("[T]he power of the system may be shifted toward the subsystem having a higher workload. Thus, higher power draw is allocated to and allowed for the subsystem having a higher workload.")).

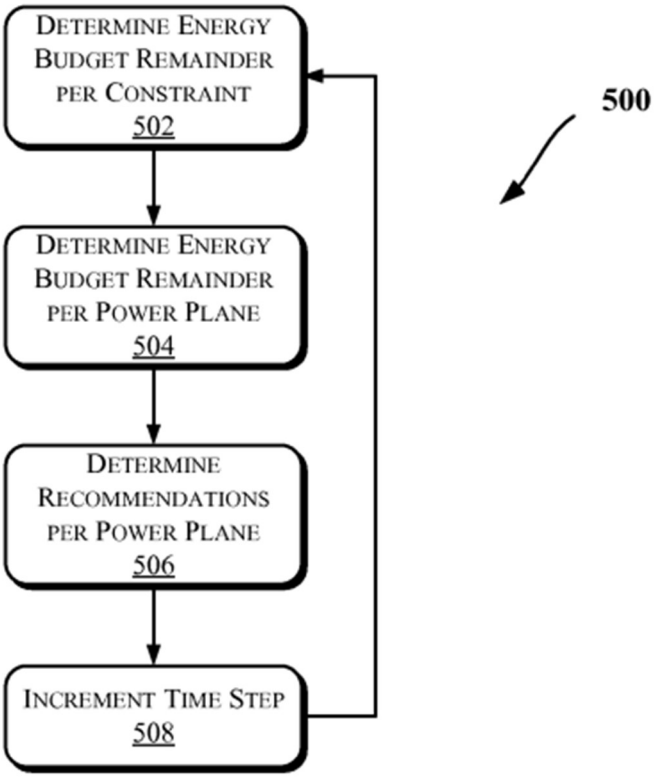
Thus, a POSITA would have at least found it obvious to increment Felter's predicted utilization ($UTIL_N$), and hence its first and second sharing policy values ($PACT_N/PACT_{TOT}$, which are proportional to $UTIL_N$), when requests for a higher

frequency on the first and second domains (first and second components, N=1, N=2) are not granted (throttled). Annavaram ¶97.

VI. Ground 2: Finkelstein Renders Obvious Claims 13, 14, and 17

A. Overview of Finkelstein (Ex-1005)

Finkelstein, published on May 6, 2010, is prior art under §102(b). Like the '316 patent, Finkelstein relates to "power management [techniques] for multiple processor cores." Finkelstein, [0001]. Figure 5 below illustrates such a technique. *See* Finkelstein, [0026]-[0040].



Finkelstein, FIG. 5.

Finkelstein teaches to implement such power management in a computer system like system 100 below. Finkelstein, [0010], FIG. 1.

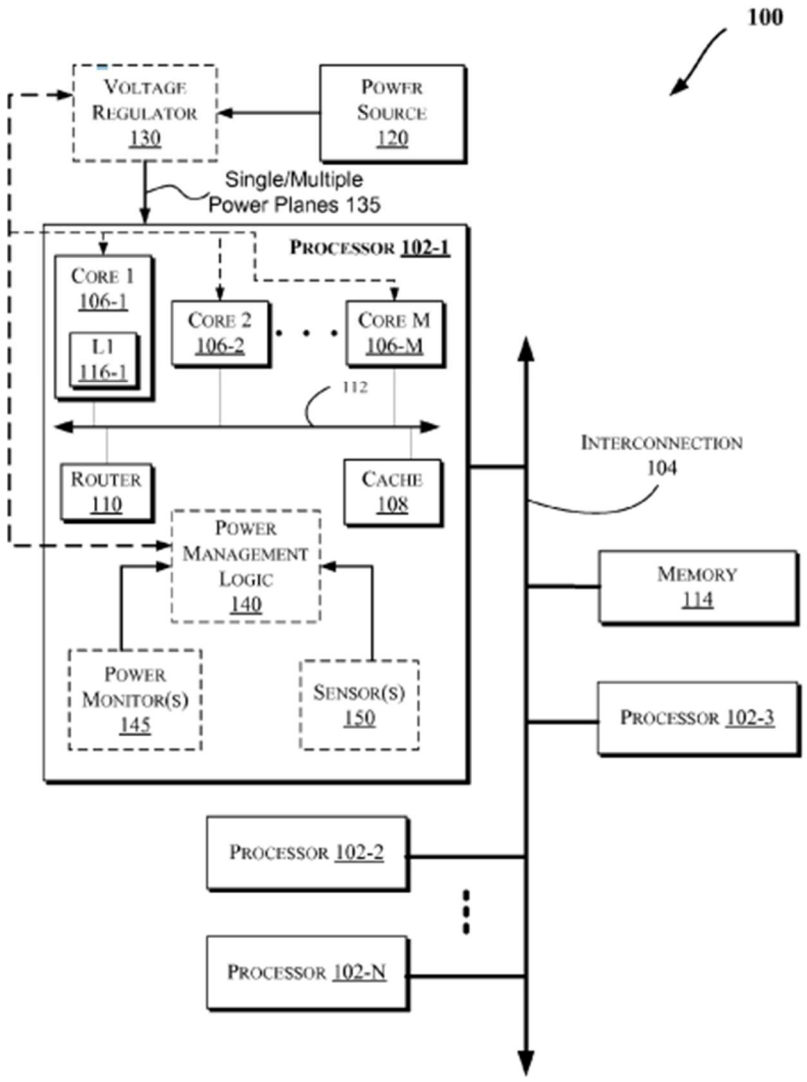


FIG. 1

Finkelstein, FIG. 1.

Computing system 100 includes a multicore processor 102-1 having "one or more processor cores 106-1 through 106-M," power source 120, voltage regulators (VR) 130, and power management logic 140. Finkelstein, [0010]-[0011], [0014]-[0016]. "[T]he voltage regulator(s) 130 may be coupled to a single power plane 135 (e.g., supplying power to all the cores 106) or to multiple power planes 135 (e.g., where each power plane may supply power to a different core or

group of cores).” Finkelstein, [0014]. Finkelstein’s processor may also include a “graphics … controller[]” comprising a “graphics and memory control hub (GMCH)” with “a graphics interface 614 that communicates with a graphics accelerator 616” via “an accelerated graphics port (AGP).” Finkelstein, [0011], [0043]-[0046] (explaining that “the processor[] and the GMCH[] may be combined to form a single chip” and “the graphics accelerator 616 may be included within the GMCH 608”), FIG. 6; Annavaram ¶62, ¶101.

Power management logic 140 “control[s] supply of power” to the cores by “instruct[ing] the VR 130, power source 120, and/or individual components of system 100 (such as the cores 106) to modify their operations.” Finkelstein, [0009]; *see* [0014]-[0016], [0032]-[0034] (describing methods for distributing power). The processor cores and graphics engine reside on distinct “power planes” and “share a common package power/energy budget.” Finkelstein, [0011], [0026].

Finkelstein is analogous art because it is in the same field of endeavor as the ’316 patent (power allocation for microprocessors) and reasonably pertinent to a problem it addresses (ensuring sufficient power). *Compare* Finkelstein, [0001]-[0003], [0009], *with* Ex-1001, Abstract, 1:18-22; *see* Annavaram ¶64.

B. Claims

1. Independent Claim 13

a. [13pre]: "a system comprising:"

Finkelstein discloses [13pre] because it discloses a "computing system" like the one below. Finkelstein, [0010]-[0016], [0042]-[0047], FIGS. 1, 6; Annavaram ¶98.

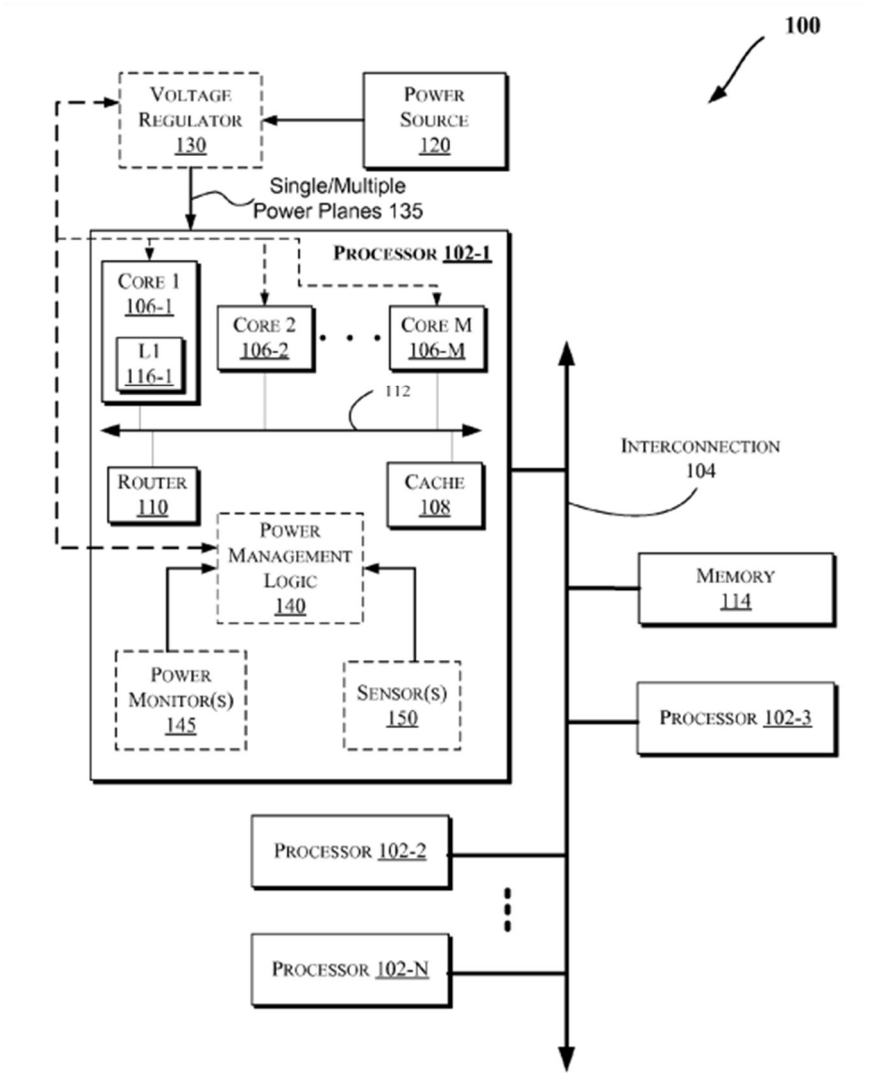


FIG. 1

Finkelstein, FIG. 1.

b. [13a]: "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,"

Finkelstein discloses or renders obvious [13a] because computing system 100 comprises a multicore, "multiple power plane" processor 102 including (1) a first power plane (domain) having first and second cores 106-1 and 106-2, (2) a second power plane (domain) including a "graphics … controller[]" (graphics engine) for "Graphics Effect(s) (GFX)," and (3) a third power plane (domain) including "power management logic" (system agent circuitry). Finkelstein, [0011], [0014], [0026], [0046], FIGS. 1, 6; Annavaram ¶¶99-105.

Finkelstein discloses a first power plane (domain) including multiple processor cores. Finkelstein's processor includes "multiple power planes 135," where "each power plane may supply power to a different core or ***group of cores***." Finkelstein, [0014] (emphasis added); *see* [0009], [0011], [0026], FIG. 1. Each power plane "may have [a] corresponding voltage regulator[]," such it operates at an independent voltage. Finkelstein, [0014]. Thus, a POSITA would have understood each power plane corresponds to a domain. Finkelstein, [0009], [0014]; Ex-1001, 4:2-5 (equating a "domain" with a "plane"); Annavaram ¶100. One power plane may supply "processor core power" for a "group of cores," whereas another may power a "graphics … controller[]," as explained below. Finkelstein, [0014], [0026], [0043].

Finkelstein discloses a second domain including a graphics engine. Finkelstein discloses a "graphics … controller[]" (second domain including a graphics engine) comprising a "graphics and memory control hub (GMCH)" with "a graphics interface 614 that communicates with a graphics accelerator 616" via "an accelerated graphics port (AGP)." Finkelstein, [0011], [0044]-[0046] (explaining "graphics accelerator 616 may be included within the GMCH 608" and "the processor[] and the GMCH[] may be combined to form a single chip"), FIG. 6; *see* Lalonde, [0022]-[0027], FIG. 5; Annavaram ¶101.

The processor cores (first domain) and graphics engine (second domain) may "share a common package power/energy budget" such that "unused processor core power" may be used "for more Graphics Effect(s) (GFX) performance in a GFX intensive workload, when processor resources are not fully utilized." Finkelstein, [0011], [0026]. This matches the '316 patent, which explains the graphics engine performs "a graphics intensive workload." Ex-1001, 6:23-26; Annavaram ¶101. Finkelstein's second domain includes the features highlighted below.

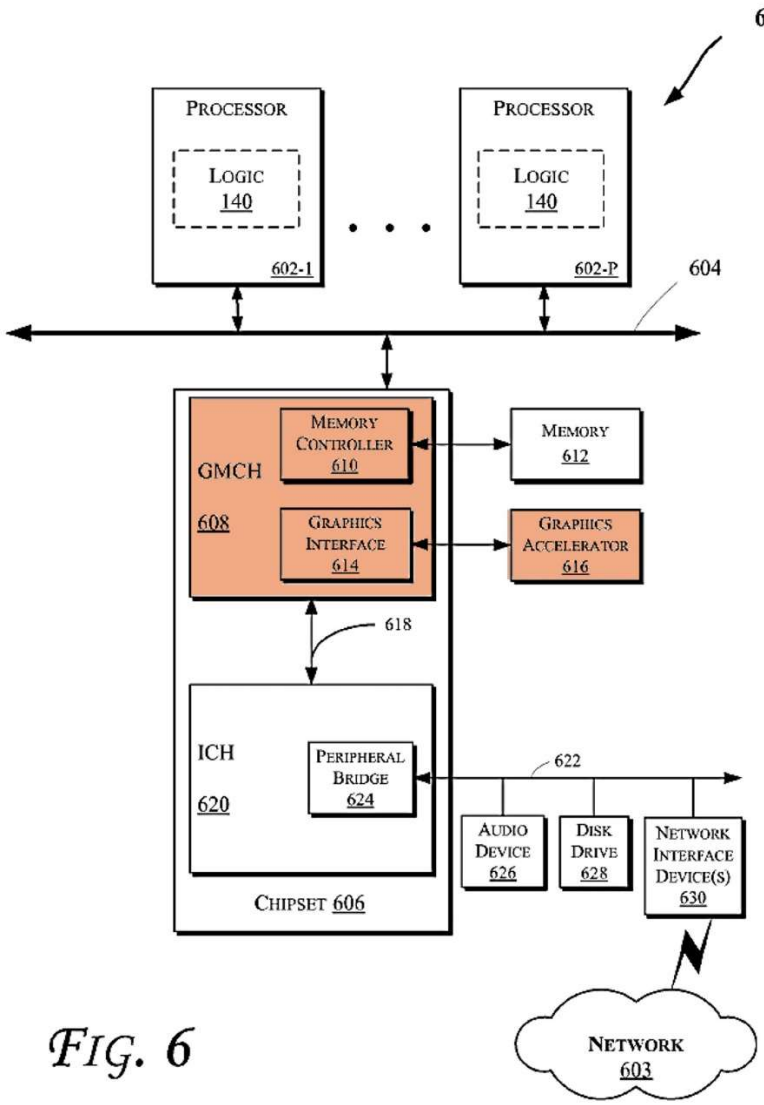


FIG. 6

Finkelstein, FIG. 6 (annotated).

Finkelstein discloses or suggests a third domain including system agent circuitry. Finkelstein's third domain, like the "system agent domain" of the '316 patent, includes power management logic 140 (system agent circuitry) to manage power for multiple power planes. *See* Finkelstein, [0014], [0016], [0023], [0028]-[0040]; Ex-1001, 9:10-16 (explaining the system agent domain "handle[s] power control events and power management"); Annavaram ¶102.

Finkelstein explains the power management logic "control[s] supply of power to components of the processor" and may reside in the processor (as Figure 1 shows), the voltage regulator, or the power supply. Finkelstein, [0016]-[0017]. Because it controls power allocation to other components of the processor, a POSITA would have understood Finkelstein's power management logic would not be allocated power dynamically; it would operate on an independent power plane with a fixed budget so it can reliably allocate power to variable-use components. *See* Finkelstein, [0014], [0016]-[0017], [0039]; Annavaram ¶103. Thus, Finkelstein's power management logic 140 operates on a third domain within the meaning of the '316 patent, as further described below regarding [13b]. Annavaram ¶103.

Annotated Figure 1 below illustrates Finkelstein's multicore, "multiple plane" processor 102-1, including a first domain comprising a group of cores 106-1 and 106-2[4] (*see* Finkelstein, [0014]) and a third domain including system agent circuitry (*see* Finkelstein, [0039]). The system agent circuitry includes power sharing logic to determine a variable power budget for a current time interval, as explained below for [13c]. Annavaram ¶104.

[4] Cores 106-1 and 106-2 are illustrative. Any plurality of grouped cores would match the claimed first domain.

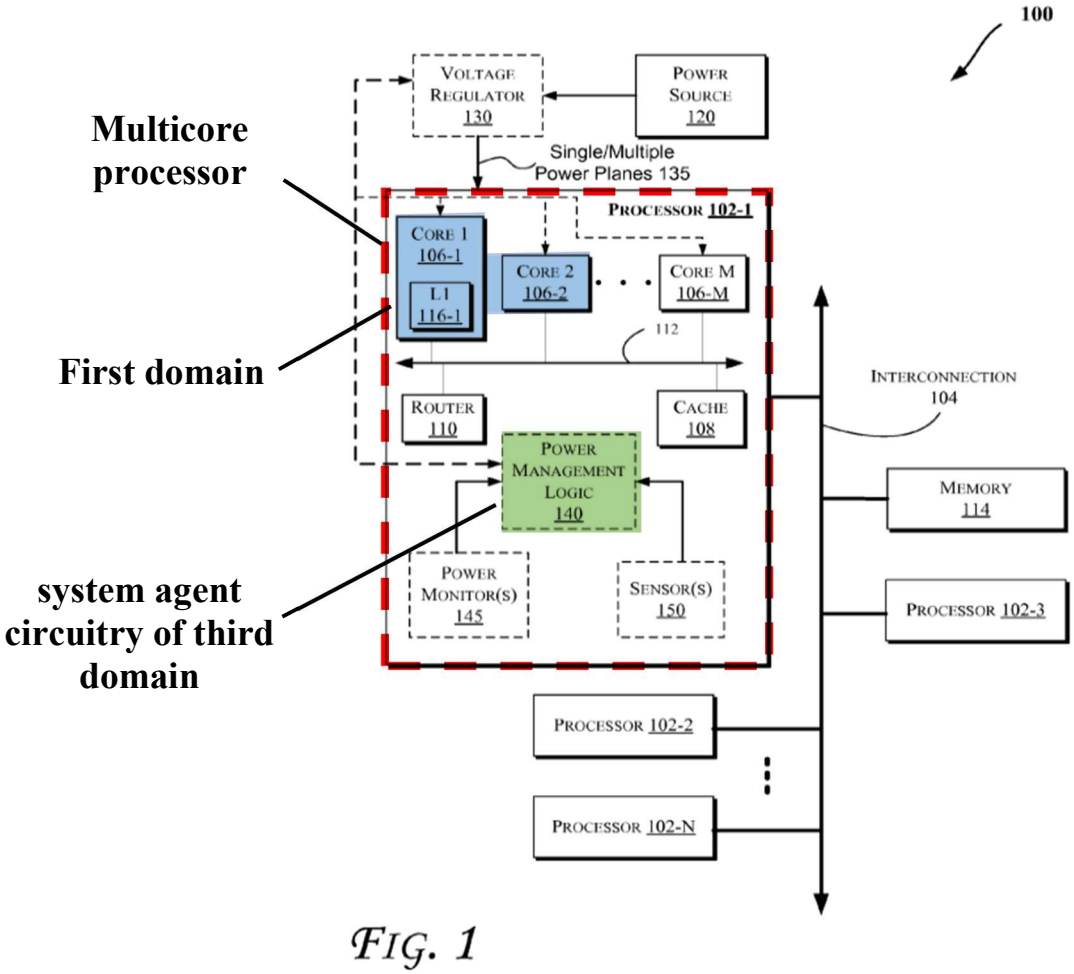


FIG. 1

Finkelstein, FIG. 1 (annotated).

Annotated Figure 6 below further illustrates a second domain for "Graphics Effect(s) (GFX) performance" (*see* Finkelstein, [0026]) that may be incorporated in processor 102-1 of Figure 1. Finkelstein, [0011], [0046]; *see* Lalonde, [0022]-[0027], FIG. 5; Annavaram ¶105.

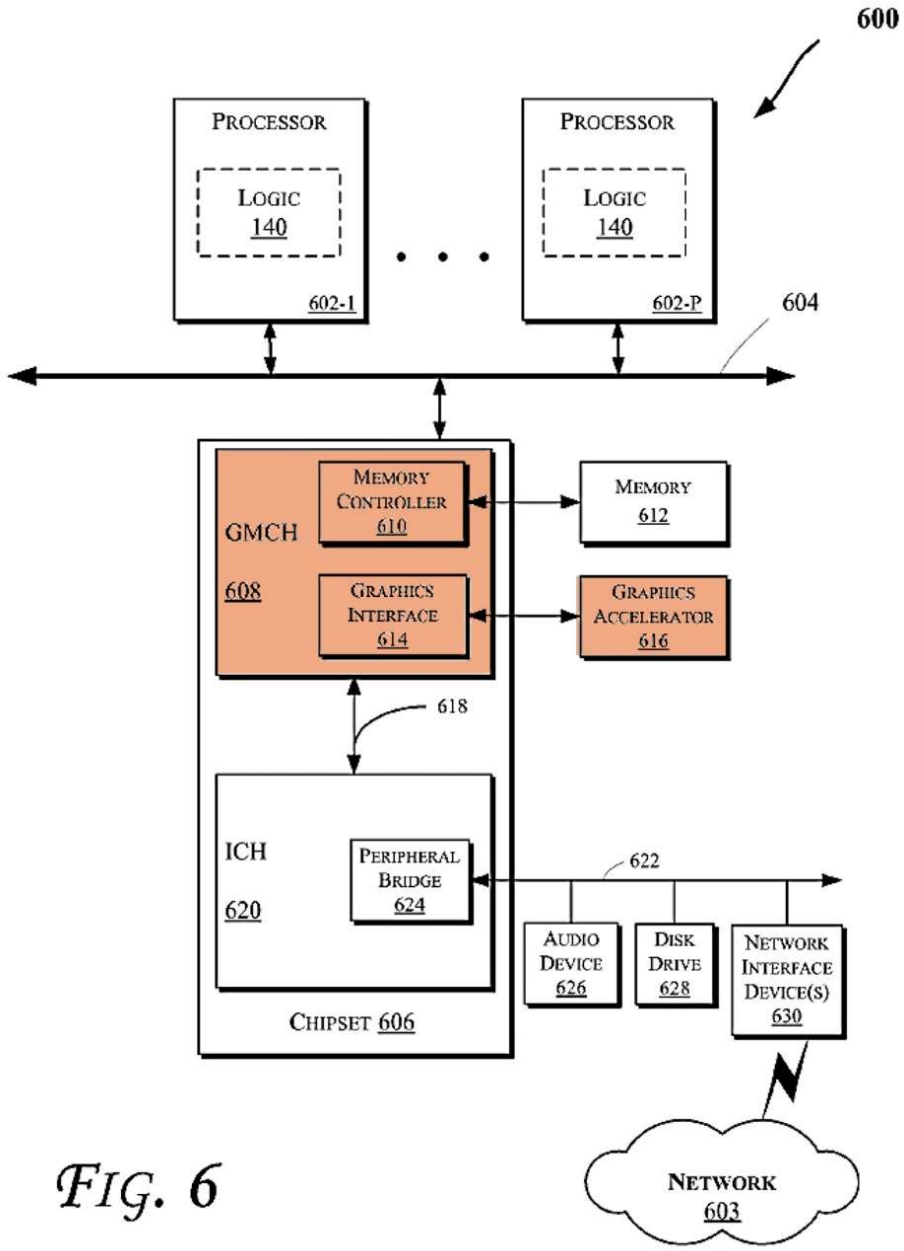


FIG. 6

Finkelstein, FIG. 6 (annotated).

c. [13b]: "the third domain to operate at a fixed power budget and to dynamically allocate a variable power budget between the first and second domains, wherein"

Finkelstein discloses or renders obvious [13b] because the third domain (power management logic) operates on a fixed power budget and dynamically

allocates a variable power/energy budget (E^k)[5] between the first domain (processor power plane) and second domain (graphics power plane) using a pro rata technique. Finkelstein, [0034]; *see* Abstract, [0009], [0026]-[0028]; Annavaram ¶¶106-114.

Finkelstein's third domain dynamically allocates the variable budget between the first and second domains. The power management logic receives information about constraints affecting energy consumption. It then uses those constraints to determine a power/energy budget (E^k) per constraint (k), which may be a package budget shared by the processor and graphics planes. *See* Finkelstein, [0024], [0026]-[0027], [0034], FIG. 5; Annavaram ¶¶107-108. For each power plane i subject to constraint k, "user (or application defined) preferences" denoted by "vectors W_k" determine how to allocate the power budget E^k "among power planes." Finkelstein, [0034]; Annavaram ¶108. A portion ($E^{k,i}$) of the power budget (E^k) is allocated pro

[5] Although Finkelstein expresses the budget as an energy in Joules, a POSITA would have understood the claimed "power budget" contemplates either an energy budget in Joules or a power budget in Watts. *See* Ex-1001, 3:22-62 (using power and energy interchangeably), claim 7 (same). This is because a power budget can be converted to an energy budget, and vice versa, by multiplying the power budget by the time interval (or dividing the energy budget by the time interval). *See id.*; Annavaram ¶51, ¶108 n.2; Finkelstein, [0027]-[0029].

rata to power plane i: "$E^{k,i}=W_{ki}E^{k}$." Finkelstein, [0034]; *see* [0039] ("[T]he energy budget remainder per power plane $E_n^{k,i}$ may be determined (e.g., by the logic 140 in accordance with formula (2) above)."); Annavaram ¶108. In this way, the power management logic (third domain) dynamically allocates the power budget between the first and second domains (i=1, i=2). Annavaram ¶109.

A POSITA would have understood the power management logic operates on a fixed power budget because it "control[s] supply of power to components of the processor" and would have required a steady stream of power (fixed power budget) to ensure system performance. *See* Finkelstein, [0016], [0039]; Annavaram ¶110. A POSITA would have understood using a fixed power budget ensures the power management logic senses "operating frequency, operating voltage, power consumption, [and] inter-core communication activity" and "instruct[s] the VR 130, power source 120, and/or individual components of system 100 (such as the cores 106) to modify their operations." Finkelstein, [0016], [0024], [0031]; *see* Annavaram ¶110.

To the extent Patent Owner disputes or the Board finds that Finkelstein does not at least suggest Finkelstein's power management logic operates on a fixed power budget, a POSITA would have been motivated to configure it to operate on a fixed power budget to ensure it consistently and reliably manages power budgets. Annavaram ¶¶110-114. Indeed, it was routine in the art to operate such logic (e.g., a

microcontroller) on a fixed power budget (i.e., as a "non-throttled" component) to dynamically budget power to variable-workload "throttled" components, including "central processing" and "graphics processing" units like the ones Finkelstein discloses. Annavaram ¶113 (citing Conroy, [0081], [0094], [0203]; Felter, [0030]).

A POSITA also would have reasonably expected to succeed because such a configuration would have required only routine skill to implement. Annavaram ¶114. For example, a POSITA would have known to configure the logic (e.g., using lookup tables or knowledge of system performance) to operate a fixed power so it can manage "how budget E^k is distributed among power planes." Finkelstein, [0034]; *see* Felter, [0054]; Annavaram ¶114 (citing Conroy, [0146], [0165]). Again, it was standard to operate such power management logic on a fixed power budget. *See* Annavaram ¶114 (citing Conroy, [0081], [0094] (describing a "non-throttled" microcontroller that "dynamically" budgets "power usage" across "Central Processing Unit (CPU)" and "Graphical Processing Unit (GPU)" domains)).

Accordingly, a POSITA at least would have found it obvious to operate Finkelstein's power management logic on a fixed power budget. Annavaram ¶¶106-114.

d. [13c]: "the system agent circuitry includes a power sharing logic to determine the variable power budget for a current time interval and"

Finkelstein discloses [13c] because its power management logic (power sharing logic) implements a variable power budget "defined iteratively" from one time to the next using formula (1). Finkelstein, [0028]-[0029]; *see* [0016], [0026]-[0040] ("[T]he energy budget remainder per constraint E_n^k may be determined (e.g., by the logic 140 in accordance with formula (1) above), FIG. 5; Annavaram ¶115. That is, Finkelstein discloses a power sharing logic (power management logic) for determining the variable power budget (E^k_{n+1}) for a current time interval (n+1). *See* Finkelstein, [0028]-[0029] ("$E_{n+1}=\alpha E_n+(TDP_n-P_n)\Delta t_n$"), [0039]; Annavaram ¶116.

e. [13d]: "to allocate a first portion of the variable power budget to the first domain according to a first power sharing value for the first domain, and to allocate a second portion of the variable power budget to the second domain according to a second power sharing value for the second domain; and"

Finkelstein discloses [13d] because its power management logic distributes first and second portions ($E^{k,i}$) of the budget (E^k) to the first and second domains (processor and graphics power planes, respectively) according to their respective power sharing values (W_{ki}). Annavaram ¶¶117-119.

As explained for [13b], the power management logic (system agent circuitry) applies "user (or application defined) preferences" to determine a portion ($E^{k,i}$) of the power budget (E^k) to distribute to each power plane i. Finkelstein, [0034], [0039];

Annavaram ¶118. The portion ($E^{k,i}$) of the power budget (E^{k}) is allocated pro rata to power plane i using formula (2): "$E^{k,i}=W_{ki}E^{k}$." Finkelstein, [0034]; *see* [0039] ("[T]he energy budget remainder per power plane $E_n^{k,i}$ may be determined (e.g., by the logic 140 in accordance with formula (2) above)."). In this way, the power management logic may "satisf[y] . . . both individual constraints per power plane and global constraints per package." Finkelstein, [0026]. For example, "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." Finkelstein, [0026]; *see* Annavaram ¶118.

Accordingly, Finkelstein discloses allocating first and second portions ($E^{k,i}$) of the power budget (E^{k}) according to first and second sharing policy values (W_{ki}) for the first and second power planes (i=1, i=2). *See* Finkelstein, [0026], [0034], [0039]; Annavaram ¶119.

f. [13e]: "a dynamic random access memory (DRAM) coupled to the multicore processor."

Finkelstein discloses [13e] because its computer system includes a "memory" that "communicate[s] with the processors 102 via [an] interconnection 104." Finkelstein, [0013]; *see* FIG. 1; Annavaram ¶120. This memory may be a "dynamic RAM (DRAM)." Finkelstein, [0043]; *see* [0042]-[0043], FIG. 6.

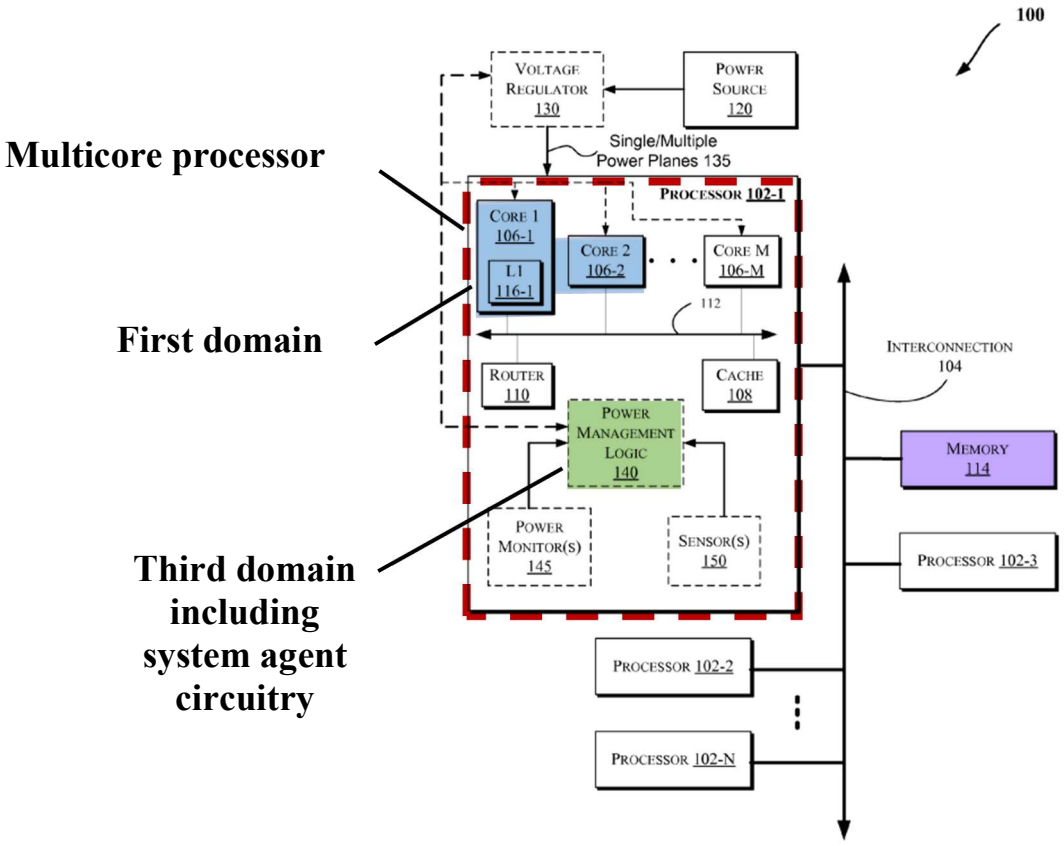


FIG. 1

Finkelstein, FIG. 1 (annotated).

2. Claim 14: "The system of claim 13, 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."

Finkelstein discloses or renders obvious claim 14 because its power management logic distributes the variable power budget in a "temporal" manner that accounts for each domain's workload for a prescribed time interval, including by allocating the "entire budget" to a "single power plane" when needed. *See* Finkelstein, [0026]-[0028], [0032]-[0034]; Annavaram ¶¶121-130.

Finkelstein's power management logic "continuously" or "periodic[ally]" determines a portion of the power/energy budget to allocate each power plane for a given "time period." Finkelstein, [0025], [0039]-[0040]; Annavaram ¶122. The allocation distributes the budget "among [the] power planes" from one time interval to the next on a pro rata basis. Finkelstein, [0034]; *see* [0039]-[0040]. Thus, a POSITA would have understood Finkelstein's power management logic dynamically allocates the variable power budget (A) to the first and second domains based on a first workload that occurs during a first time period and (B) to the first and second domains based on a second workload executed after the first load (during a subsequent time period). *See* Annavaram ¶122; Finkelstein, [0026], [0039]-[0040].

Finkelstein at least suggests allocating substantially all the variable power budget to the first domain for a first workload and then to the second domain for a second workload. Finkelstein teaches to share a "***common package power/energy budget***" between the processing and graphics power planes and explains "power management may need to be satisfied … ***per package***." Finkelstein, [0026] (emphases added). According to Finkelstein, "the entire" shared package power budget may be allocated to one of the power planes during a prescribed time period. Finkelstein, [0034]; *see* Annavaram ¶123. For example, "a ***single power plane*** may obtain ***the entire budget***" based on "user (or application defined) preferences that describe how budget E^k is distributed." Finkelstein, [0034]. Finkelstein thus at least

suggests allocating substantially all the power budget to one power plane (e.g., processor cores) when the "user (or application defined) preferences" W_{ki} require as much for a first workload. Annavaram ¶123.

Finkelstein also explains "unused processor core power" may be redistributed to the graphics controller "for more Graphics Effect(s) (GFX) performance." Finkelstein [0026]. A POSITA would have understood the processor cores (first domain) would require little power during a graphics-heavy workload, so substantially all of the power budget should be allocated to the graphics controller (second domain). Likewise, for a processor-heavy workload, substantially all the power budget should be allocated to the processor cores, not the graphics controller. Annavaram ¶124.

This was well known. Annavaram ¶125 (citing, e.g., Gwennap, 6 ("If the GPU [(Graphics Processing Unit)] is not in use or is operating at a low speed, ***the power manager can assign some of its thermal budget to the CPUs***, allowing them to run at a higher clock speed. Conversely, ***on a graphics-intensive application, the power manager can overclock the GPU*** while slowing down the CPUs." (emphasis added)). For example, a POSITA would have understood that many applications, like program development tools and scientific applications, produce a heavy load for the processor cores and almost no load for the graphics controller. Annavaram ¶126 (citing Conroy [0196]). Other applications, like advanced user interfaces or graphics

editing applications, present the opposite situation. *See id*. (citing Conroy [0196]). Accordingly, "[m]any professional applications present an alternating high workload to the CPU and to the GPU." Conroy [0196].

Knowing this, a POSITA would have expected a multi-domain processing system like Finkelstein's to alternate between executing processor-heavy workloads and graphics-heavy workloads, allocating substantially all the power budget to the processor cores for processor-heavy workloads and substantially all the power budget to the graphics controller for graphics-heavy workloads. *See* Finkelstein, [0026], [0034]; Annavaram ¶127.

To the extent Patent Owner disputes this or the Board determines Finkelstein does not disclose allocating substantially all the power budget to the second domain for a second workload after the first, a POSITA would have been motivated to do so. For example, a POSITA would have been motivated to configure Finkelstein's power management logic to allocate substantially all the available power budget to the first domain (processor plane) for a processor-heavy first workload, then allocate substantially all the available power budget to the second domain (graphics controller plane) for a graphics-heavy second workload. Annavaram ¶128 (citing Gwennap, 6; Conroy [0196]).This would have ensured optimal performance. *Id.*

A POSITA would have reasonably expected to succeed at this because it only would have required routine skill. Annavaram ¶129. By setting the power

distribution preferences (W_{ki}) accordingly, *see* Finkelstein, [0034], a POSITA would have known how to configure the power management logic to allocate substantially all the available power to the first domain for first (processor-heavy) workload, and subsequently to allocate substantially all the available power to the second domain for a second (graphics-heavy) workload. Annavaram ¶129 (citing Therien, [0014]; Conroy, [0191], [0195]-[0196]).

Accordingly, a POSITA would have found it obvious to configure Finkelstein's power management logic to prioritize the first or second domains for a first or second workload such that substantially all the available power budget is allocated to the first domain for a first (processor-heavy) workload and then to the second domain for a second (graphics-heavy) workload. Annavaram ¶130.

3. Claim 17: "The system of claim 13, 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."

Finkelstein at least suggests claim 17 because it discloses prioritizing power distribution to the "GFX intensive workload, when processor resources are not fully utilized" and allocating power to a "single power plane" when needed or as "user (or application defined) ***preferences***" dictate. Finkelstein, [0026]-[0028], [0032]-[0034] (emphasis added); Annavaram ¶¶131-136.

By applying the power sharing methods described for claim 13, Finkelstein explains "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." Finkelstein, [0026]; *see* [0050] (describing GFX functions as "high-performance"). To adequately support GFX functionality, a POSITA would have understood high-performance workloads performed on the graphics controller's power plane (second domain) would have been prioritized / preferred, at least for graphics-heavy workloads, over the processor power plane (first domain). Annavaram ¶132.

This was well known. *See* Annavaram ¶133 (citing Gwennap, 6 ("If the GPU [(Graphics Processing Unit)] is not in use or is operating at a low speed, the power manager can assign some of its thermal budget to the CPUs, allowing them to run at a higher clock speed. Conversely, ***on a graphics-intensive application, the power manager can overclock the GPU while slowing down the CPUs***." (emphasis added)); Therien, [0017] ("[A] computing system might be configured such that ***a graphics subsystem will typically receive preferential power allocations as compared to a CPU***." (emphasis added)); Bose, Abstract, [0010], [0021], [0085] (describing a "thread scheduler" that "priorit[izes]" and "assigns" power distribution to "high performance threads" over "low performance" threads)). Accordingly, a POSITA would have understood when the graphics power plane (second domain) is

preferred for a given time interval, a preference value favors it (e.g., $W_{k,2}>W_{k,1}$). *See* Finkelstein, [0034]; Annavaram ¶134-136.

To the extent Patent Owner disputes this or the Board determines otherwise, a POSITA would have been motivated to allocate Finkelstein's available variable power budget to the graphics controller domain according to a preference value to ensure graphics performance, as was known in the art. *See* Annavaram ¶135 (citing Gwennap, 6; Therien, [0017]; Bose, [0010], [0021], [0085]).

A POSITA would have reasonably expected to succeed in doing so because it only would have required routine skill to implement. Annavaram ¶136. For example, because the power distribution preferences may be user or application defined, a POSITA would have known to program Finkelstein's power management logic to favor the second domain over the first (e.g., setting $W_{k,2}>W_{k,1}$). Annavaram ¶136 (citing Therien, [0014]; Conroy, [0191], [0195]-[0196]).

VII. Ground 3: The Finkelstein-Felter Combination Renders Obvious Claims 8 and 12-17

A. A POSITA Would Have Found It Obvious to Adapt Finkelstein's Pro Rata Power Management Logic to Perform Felter's "Modified Pro Rata Technique" for Power Allocation

Finkelstein and Felter describe very similar techniques for managing power in similar systems. *See* Annavaram ¶¶137-150. For example, both references similarly define power budgets. Because "very recent activity is a good predictor of current activity," Felter teaches to budget power "for time period N+1 [using] a

simple function of the monitored activity for time period N." Felter, [0051]. Finkelstein's formula (1) does exactly that. *See* Finkelstein, [0028]-[0029] ("$E_{n+1}=\alpha E_n+(TDP_n-P_n)\Delta t_n$").

Both references also similarly allocate power budgets. As explained for claim 13, Finkelstein allocates power on a pro rata basis:

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

where E^k is the power budget, and W_{ki} represent "user (or application defined) preferences." Finkelstein, [0034]. Felter similarly allocates power "us[ing] a ***modified*** pro rata technique," as explained for claim 8. Felter, [0061] (emphasis added). Like Finkelstein, Felter allocates power based on certain preferences, but only after first reserving and allocating a "minimum power" for each domain:

$$\mathbf{PALLOC_N = PMIN_N + (PACT_N/PACT_{TOT})\ (P_{BUDGET} - SUM(PMIN_N)),}$$

where $\mathbf{PMIN_N}$ are the minimum powers, $\mathbf{P_{BUDGET}}$ is the power budget, and $\mathbf{PACT_N/PACT_{TOT}}$ allocates the remaining power. Felter, [0057]-[0061], FIG. 5.

According to Felter, this modified pro rata technique "effectively distributes the available system power to the system components in a manner that (1) ensures that each component receives a minimum level of power and (2) allocates the available active power in proportion to the requirement for the components by the workload(s) executing on the system." Felter, [0061]. Moreover, including "a reservoir for minimal component activity" in the standby power allocated to each

component "ensure[s] there is always enough power for each component to obtain some minimal work from it." *Id.*

For the same reasons, a POSITA would have found it obvious to adapt Finkelstein's power management logic to apply Felter's modified pro rata technique. That is, a POSITA would have found it obvious to adapt Finkelstein's formula (2) according to Felter's power allocation equations in Figure 5:

$$E^{k,i} = E^{k,i}_{min} + W_{k,i}\left(E^{k} - \sum_i E^{k,i}_{min}\right) \qquad \textbf{(Eq. 2, modified)}$$

where $E^{k,i}{}_{min}$ are the minimum powers, E^{k} is the power budget, and $W_{k,i}$ allocates power based on activity as in Felter. *Compare* Felter, [0051], [0057]-[0061], *with* Finkelstein, [0028]-[0029], [0034]; *see* Annavaram ¶¶142-145.

First, a POSITA would have been motivated to configure Finkelstein's power management logic to include a minimum power ($E^{k,i}{}_{min}$) for each power plane, thereby ensuring each component has "enough power" to perform "minimal work." Felter, [0065]; Annavaram ¶146. Such a modification would have been particularly useful to ensure "smooth operation of critical applications." Finkelstein, [0024]; Annavaram ¶146.

Second, a POSITA would have been motivated to configure Finkelstein's power management logic to allocate the available active power based on each domain's needs according to "the workload(s) executing on the system." Felter, [0061]. This approach "distributes scarce system power equitably so that the system

performance of each component is impacted proportionately." Felter, [0061]. A POSITA would have found this to be advantageous when applied to Finkelstein's disclosures. *See* Finkelstein, [0026]; Annavaram ¶¶147-148.

For example, Finkelstein teaches "user (or application defined) preferences that describe how budget E^k is distributed among power planes." Finkelstein, [0034]. Although Finkelstein does not provide examples, a POSITA would have recognized that Felter's modified pro rata technique provides one beneficial way to do so. Annavaram ¶148. According to Felter, "the active power predicted for a component N is indicated as the product of the activity (UTIL) predicted for component N ($UTIL_N$) and a constant of proportionality for component N (C_N)." Felter, [0059], FIG. 5. A POSITA would have found it obvious for either a user or application to set the values C_N according to efficiency/performance preferences. A POSITA also would have understood applications dictate the predicted activity ($UTIL_N$). *See* Finkelstein, [0026] (explaining the graphics domain may require more power under "a GFX intensive workload"); Annavaram ¶149 (citing Gwennap, 6 ("If the GPU [(Graphics Processing Unit)] is not in use or is operating at a low speed, the power manager can assign some of its thermal budget to the CPUs, allowing them to run at a higher clock speed. Conversely, on a graphics-intensive application, the power manager can overclock the GPU while slowing down the CPUs.")), ¶149 (citing Conroy [0196] ("Many professional applications present an alternating high

workload to the CPU and to the GPU that results in an alternating asymmetric load profile of the system."). Accordingly, a POSITA would have found it obvious to set Finkelstein's "user (or application defined) preferences" ($W_{k,i}$) by applying Felter's teachings. In fact, Finkelstein's $W_{k,i}$ values are mathematically analogous to Felter's values $PACT_N/PACT_{TOT}$, which are set by $UTIL_N$ and C_N. Annavaram ¶149; *see* Felter, [0059], FIG. 5.

When adapting Finkelstein's power management logic to perform Felter's modified pro rata power allocation technique, a POSITA would have reasonably expected to succeed because it would have required only routine skill to implement. Annavaram ¶149. Again, $W_{k,i}$ and $PACT_N/PACT_{TOT}$ are mathematically equivalent for purposes of allocation. *Id.* And Felter's method of determining $PACT_N$ ($PACT_N=UTIL_N*C_N$) reflects a simple way of defining Finkelstein's "user (or application defined) preferences" $W_{k,i}$. Finkelstein, [0034]; *see* Felter, [0059], FIG. 5; Annavaram ¶149.

Moreover, the minimum power values ($E^{k,i}_{min}$) would have required only routine skill to acquire, e.g., through manufacturer guidelines, routine observation of each power plane's power consumption, or discrete settings. *See* Annavaram ¶150 (citing Bose, [0048], [0055]); Felter, [0009], [0039], [0058]; Finkelstein, [0034]. A POSITA thus would have reasonably expected to succeed when adapting

Finkelstein's power allocation logic to perform Felter's modified pro rata technique. Annavaram ¶150.

Because a POSITA would have been motivated to adapt Finkelstein's power management logic to perform Felter's modified pro rata power allocation technique and would have reasonably expected to succeed, a POSITA would have found it obvious. Annavaram ¶¶138-150.

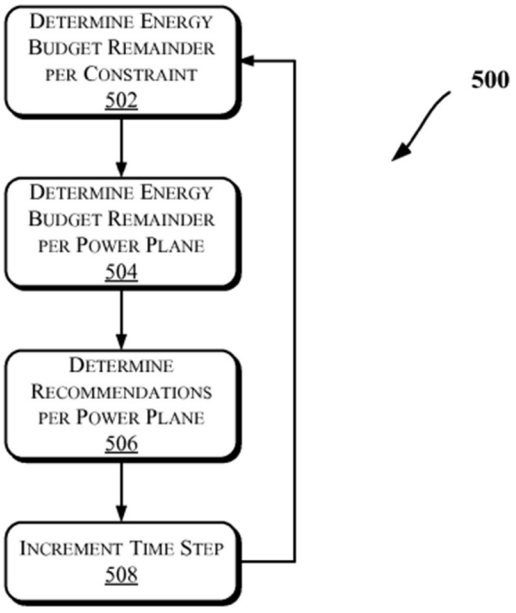
The analysis below applies this combination to demonstrate why claims 8 and 12-17 are unpatentable.

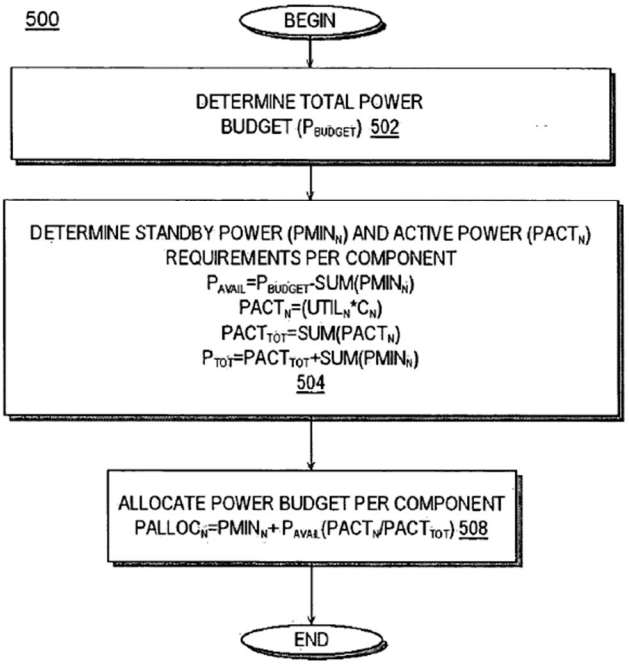
B. Claims

1. Independent Claim 8

a. [8pre] (method)

The Finkelstein-Felter combination satisfies [8pre]. Annavaram ¶¶151-152. As Section VII.A explains, Finkelstein and Felter disclose similar methods for allocating power in similar multicore processor systems. Finkelstein discloses a method for "provid[ing] efficient power management for multiple processor cores." Finkelstein, Abstract, [0009]; *see* FIG. 5 (below, left). Felter similarly discloses "a method for allocating power budgets to the system components based on expected levels of activity." Felter, [0016]; *see* [0057]-[0061], FIG. 5 (below, right).





Finkelstein, FIG. 5 Felter, FIG. 5

In the Finkelstein-Felter combination, Finkelstein's power management logic is adapted to perform Felter's modified pro rata power management technique. *See supra* §VII.A.

b. [8a] (power controller determining a power budget for first and second domains)

The Finkelstein-Felter combination satisfies [8a]. Annavaram ¶¶153-159. Finkelstein discloses a "multiple core processor" (multi-domain processor) with "one or more processor cores" operating on "multiple power planes" (first and second domains). Finkelstein, [0009]. Additionally, Finkelstein discloses "power management logic" (power controller) to "define[]" and "manage[]" a "controller budget" (also called a "power/energy budget"). Finkelstein, [0009], [0016], [0028];

see [0039], FIG. 1. Finkelstein's Figure 1 below illustrates a multi-processor computing system 100 with processors 102-1 through 102-N. *See* Finkelstein, [0010], FIG. 1.

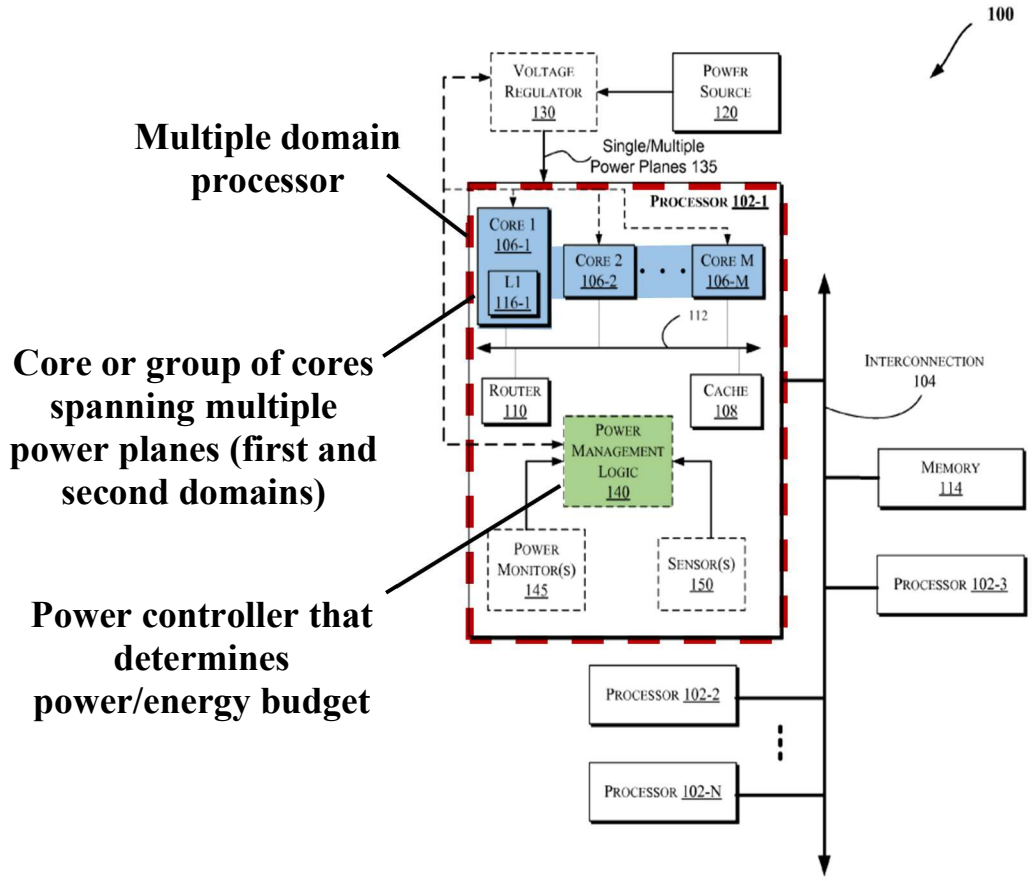


FIG. 1

Finkelstein, FIG. 1 (annotated).

The Finkelstein-Felter combination employs Finkelstein's computing system configured as described in Section VII.A. Each processor 102 has multiple cores 106-1 through 106-M, and power management logic 140 "coupled to" voltage regulator(s) (VR) 130. *See* Finkelstein, [0011], [0016], FIG. 1. Voltage regulators 130 regulate "***multiple power planes 135 (e.g., where each power plane may supply***

power to a different core or group of cores)." Finkelstein, [0014] (emphasis added), FIG. 1. Moreover, the power management logic prescribes a power/energy budget and determines how to allocate it among various power planes. *See* Finkelstein, [0039]. It then applies "DVS, pure throttling techniques, or some combinations thereof to control the power consumption of processors 102." Finkelstein, [0024]; *see* [0019]-[0020]. A POSITA thus would have understood each power plane corresponds to a "domain." *See* Ex-1001, 1:52-56 (defining "domain"), 4:2-5 (equating "domain" and "plane"); *supra* §VI.B.1.b ([13a]); Annavaram ¶¶154-155.

Finkelstein further explains that "power setting(s) (e.g., voltage and/or frequency changes) may be made according[] to a current budget." Finkelstein, [0028]. Finkelstein's power management logic (power controller) defines that power/energy budget (E_{n+1}) for a current time interval and allocates it among the power planes (including the first and second domains). *See* Finkelstein, [0028]-[0039]. In the Finkelstein-Felter combination, Finkelstein's power management logic is adapted to perform this task by applying Felter's modified pro rata technique. *See supra* §VII.A; Felter, FIG. 5.

The first step is to determine "a current budget" (E^k_{n+1}),[6] which is "defined iteratively" based on the power consumed during the previous interval: "$E_{n+1}=\alpha E_n+(TDP_n-P_n)\Delta t_n$." Finkelstein, [0028]-[0029] (formula (1)), [0039] ("[T]he energy budget remainder per constraint E_n^k may be determined (e.g., by the logic 140 in accordance with formula (1) above); *see* Felter, [0051], [0057], FIG. 5. In the Finkelstein-Felter combination, the power consumed is determined from domain activity. *See* Felter, [0051] ("[T]he predicted activity for time period N+1 is a simple function of the monitored activity for time period N …."), [0054] ("[P]ower administrator 200 includes allocating (block 306) power budgets to the various system components based on the predicted level of activity."); Finkelstein, [0016], [0028]-[0029]; Annavaram ¶¶156-158.

Accordingly, the Finkelstein-Felter combination determines, in a power controller (Finkelstein's power management logic, as modified by Felter) of a multi-domain processor (processor 102-1), a power budget (E^k_{n+1}) for the multi-domain

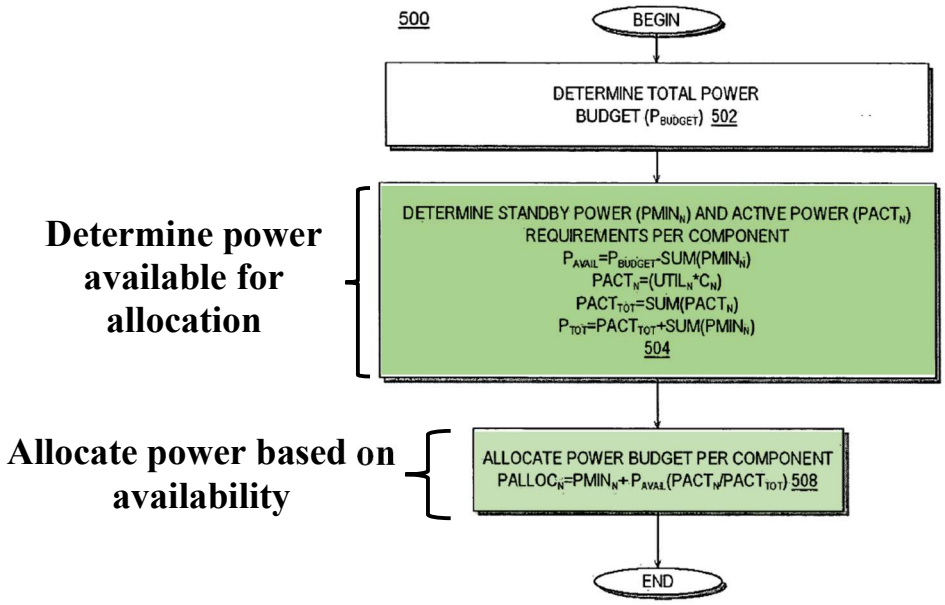
[6] Although Finkelstein allocates an energy budget and Felter allocates a power budget, a POSITA would have understood the two to be equivalent. *Supra* note 5; Annavaram ¶51, ¶155 n.4. In fact, the '316 patent equates them. *Compare* Ex-1001, 3:22-62 (explaining "E_n=energy budget" in equation [1]), *with id.*, claim 7 (explaining "E_n is the power budget" in the same equation).

processor for a current time interval (n+1), the multi-domain processor including at least a first domain (first processor power plane 135) and a second domain (second processor power plane 135). Finkelstein, [0028]-[0029], [0039]; Annavaram ¶159.

c. [8b] (determining a portion of the budget to allocate to the domains)

The Finkelstein-Felter combination satisfies [8b] because, as modified, Finkelstein's power management logic (power controller) uses Felter's modified pro rata technique to allocate the power budget to the first and second power planes. *See supra* §VII.A; Annavaram ¶¶160-164; Felter, [0050]-[0066], FIGS. 3-5.

As explained for Ground 1, Felter's power management logic sets "the amount of power budgeted for each component [$PALLOC_N$] … equal to the component's standby power requirements [$PMIN_N$] (which must always be met) and a portion [$PACT_N/PACT_{TOT}$] of the available power P_{AVAIL}." Felter, [0061].



Felter, FIG. 5 (annotated).

In the Finkelstein-Felter combination, which applies this modified pro rata technique, Finkelstein's formula (2) would be modified:

$$\boldsymbol{E^{k,i} = E^{k,i}_{min} + W_{k,i}\left(E^{k} - \sum_i E^{k,i}_{min}\right)} \qquad \textbf{(Eq. 2, modified)}$$

See supra §VII.A; Annavaram ¶¶161-162; Felter, [0061], FIG. 5; Finkelstein, [0034]. This "effectively distributes the available system power to the system components in a manner that (1) ensures that each component receives a minimum level of power and (2) allocates the available active power in proportion to the requirement for the components by the workload(s) executing on the system." Felter, [0061]; Annavaram ¶163.

Accordingly, the Finkelstein-Felter combination determines, in the power controller (Finkelstein's power management logic, as modified by Felter), a portion ($E^{k,i}$) of the power budget (E^{k}) to be allocated to the first and second domains (first and second processor core power planes, i=1 and i=2).[7] Annavaram ¶164.

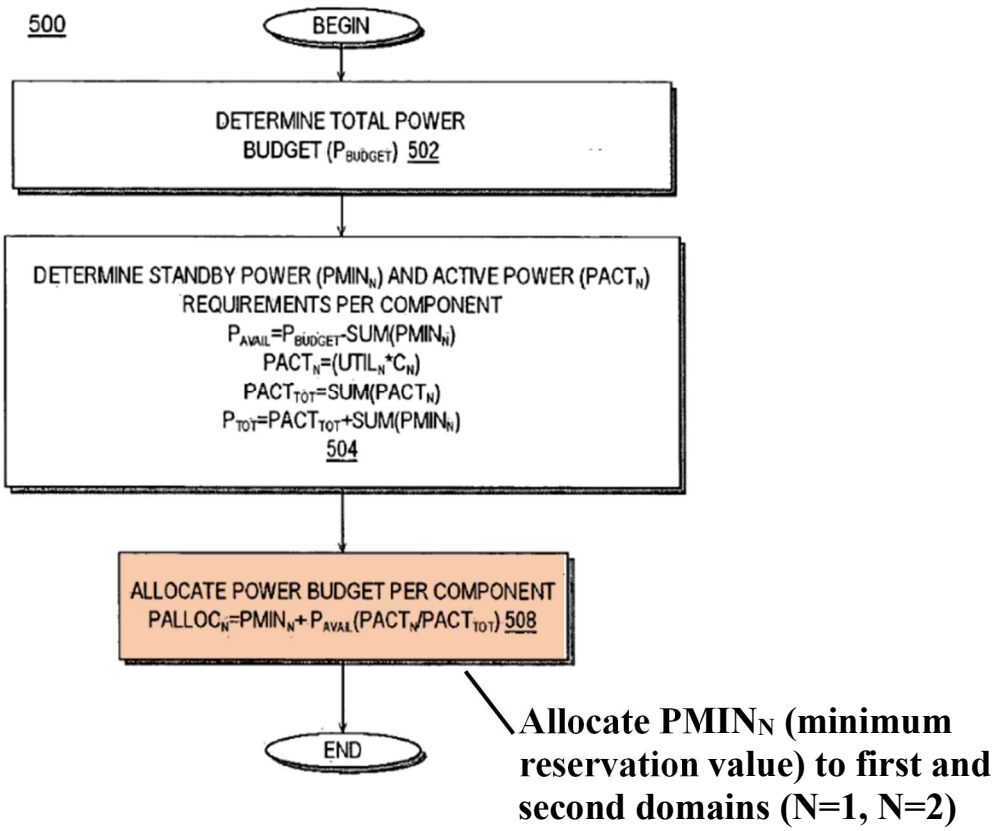
d. [8c] (allocating minimum reservation values)

In the Finkelstein-Felter combination, Finkelstein's power management logic is adapted to perform Felter's modified pro rata technique. *See supra* §VII.A. The

[7] The sum of $E^{k,i}$ over all power planes i equals the total budget E^{k}, such that $E^{k,i}$ for each power plane i corresponds to a portion of the total budget. Annavaram, ¶164 n.5.

Finkelstein-Felter combination satisfies [8c] because Felter's modified pro rata technique allocates, to each component, "the component's standby power requirements (which must always be met)" (a minimum reservation value). Felter, [0061]; *see* [0039], [0042]-[0049], [0065], FIG. 5; *see* Annavaram ¶¶165-167; *supra* §V.B.1.d (Ground 1, [8c]).

As Felter explains, each system component (including each processor core) has a "fixed" $PMIN_N$ (also called "standby power") corresponding to "the minimum operational power of the processor at its lowest operating point." Felter, [0039]; *see* [0042]-[0049], [0061], [0065]. After the power administrator determines "the power budget and the active and standby demands of each of the components" (*see* Felter, [0058]-[0060], FIG. 5 (block 504)), it allocates, to each component, "the component's standby power requirements (which must always be met) and a portion of the available power P_{AVAIL}." Felter, [0061], FIG. 5 (block 508). This "reservoir for minimal component activity" "ensure[s] there is always enough power for each component to obtain some minimal work from it." Felter, [0065]. Thus, Felter discloses allocating a minimum reservation value ($PMIN_N$) to each of the processor cores. *See supra* §V.B.1.d; Annavaram ¶166.



Felter, FIG. 5 (annotated).

In the Finkelstein-Felter combination, which applies Felter's modified pro rata technique, the same is true. Finkelstein's formula (2) is modified:

$$\boldsymbol{E^{k,i} = E^{k,i}_{min} + W_{k,i}\left(E^{k} - \sum_i E^{k,i}_{min}\right)} \qquad \textbf{(Eq. 2, modified)}$$

where $E^{k,i}{}_{\min}$ corresponds to Felter's minimum power $PMIN_N$. *See supra* §VII.A; Annavaram ¶167. Accordingly, Finkelstein's power management logic, as modified by Felter, allocates a minimum reservation value ($E^{k,i}{}_{\min}$) to the first and second power planes (first and second processor core power planes, i=1 and i=2). Annavaram ¶167.

e. [8d] (sharing a remaining portion of the budget according to sharing policy values)

In the Finkelstein-Felter combination, Finkelstein's power management logic is adapted to perform Felter's modified pro rata technique. *Supra* §VII.A. This combination satisfies [8d] because "the amount of power budgeted for ***each component*** is equal to ***the component's*** standby power requirements (which must always be met) ***and a portion of the available power***." Felter, [0057]-[0061] (emphasis added); *see supra* §V.B.1.e (Ground 1, [8d]); Annavaram ¶¶168-173.

After determining a minimum reservation value ($PMIN_N$) for each domain, Felter determines the remainder of the power budget (P_{AVAIL}): "[A]n available power figure (P_{AVAIL}) is determined by subtracting from the maximum power deliverable by the system the sum of all the standby values." Felter, [0058]; *see* FIG. 5 ("$P_{AVAIL}=P_{BUDGET} - SUM(PMIN_N)$"). P_{AVAIL} represents the claimed "remaining portion of the power budget." Annavaram ¶169.

Each domain is further allocated a portion of P_{AVAIL}: "The portion of P_{AVAIL} allocated to a system is determined by the ratio of the components active power requirements to the total active power requirements of the system. This formula is represented in block 508 as $PALLOC_N=PMIN_N+P_{AVAIL}*(PACT_N/PACT_{TOT})$." Felter, [0061]; *see* FIG. 5 (block 508). The proportions $PACT_N/PACT_{TOT}$ represent the claimed "sharing policy value[s]." Annavaram ¶¶170-171.

In the Finkelstein-Felter combination, which applies Felter's modified pro rata technique, Finkelstein's formula (2) is modified:

$$\boldsymbol{E^{k,i} = E^{k,i}_{min} + W_{k,i}\left(E^{k} - \sum_i E^{k,i}_{min}\right)} \qquad \textbf{(Eq. 2, modified)}$$

where the term in parentheses corresponds to Felter's P_{AVAIL}, and $W_{k,i}$ corresponds to Felter's $PACT_N/PACT_{TOT}$. *See supra* §VII.A; Annavaram ¶170. Like Felter's P_{AVAIL}, the term in parentheses ($E^k - \sum_i E^{k,i}_{min}$) represents a remaining portion of the power budget (E^k) after allocating the minimum power (minimum reservation value) ($E^{k,i}{}_{min}$) to each power plane (domain) i. *See* Felter, FIG. 5 ("P_{AVAIL}=P_{BUDGET}– SUM($PMIN_N$)"); Annavaram ¶171. And, like Felter's $PACT_N/PACT_{TOT}$ values, the $W_{k,i}$ values represent the sharing policy values of [8d] for each respective power plane (domain) i. *See supra* §V.B.1.e (Ground 1, [8d]), §VI.B.1.e (Ground 2, [8d]); Annavaram ¶171. In this way, Finkelstein's power management logic, as modified by Felter, "allocates the available active power [$E^k - \sum_i E^{k,i}_{min}$] in proportion to the requirement for the components by the workload(s) executing on the system." Felter, [0061]; *see* FIG. 5; Annavaram ¶171.

$PACT_N$ is "the product of the activity (UTIL) predicted for component N ($UTIL_N$) and a constant of proportionality for component N (C_N)." Felter, [0058]-[0059]; *see* FIG. 5. Because $PACT_{TOT}$ is just the sum of all $PACT_N$, Felter's sharing policy values $PACT_N/PACT_{TOT}$ ensure each domain N shares power proportional to its expected utilization. Felter, [0059]-[0061], FIG. 5. The same would be true for

the analogous $W_{k,i}$ values and power planes (domains) i in the Finkelstein-Felter combination. Annavaram ¶172.

Accordingly, Finkelstein's power management logic, as modified by Felter, shares a remaining portion ($E^k - \sum_i E_{min}^{k,i}$) of the power budget ($E^k$) according to first and second sharing policy values ($W_{k,i}$) for the first and second domains (power planes, i=1 and i=2). Annavaram ¶173.

f. [8e] (controlling the domain frequencies based on the power allocations)

In the Finkelstein-Felter combination, Finkelstein's power management logic is adapted to perform Felter's modified pro rata technique. *Supra* §VII.A. This combination satisfies [8e] because [8e] represents a fundamental purpose of dynamically allocating power, including in Finkelstein's and Felter's disclosures. Annavaram ¶¶174-176.

Finkelstein explains "power setting(s) (e.g., voltage and/or frequency changes) may be made according[] to a current budget." Finkelstein, [0028]. Power management logic 140 does this. Finkelstein, [0039]. For example, "logic 140 may request the cores 106 to ***modify their operating frequency***, power consumption, etc." Finkelstein, [0016] (emphasis added); *see* [0024] ("[L]ogic 140 may consider DVS, pure throttling techniques, or some combinations thereof to control the power consumption of processors 102."), [0019] (explaining "'pure' throttling" includes

"frequency throttling" and "dynamically adjust[ing] processor operational voltage and/or frequency"); Annavaram ¶174.

Incorporating Felter's modified pro rata power allocation technique does not change this. Annavaram ¶175. In fact, Felter uses the modified pro rata technique to do the same. Like Finkelstein, Felter regulates component (domain) activity using "voltage-frequency scaling." Felter, [0035]-[0036]; *see* [0037]-[0049] (describing voltage-frequency scaling for "two independent frequency-scalable processors"), [0065] (referencing "the voltage-frequency throttling described above"); Annavaram ¶175.

Accordingly, Finkelstein's power management logic, as modified by Felter, controls a frequency of the first domain (first processor core power plane) and a frequency of the second domain (second processor core power plane) based on the allocated portions of the power budget. Annavaram ¶176.

2. Claim 12 (the sharing policy values are incremented when requests for higher frequencies are not granted)

In the Finkelstein-Felter combination, Finkelstein's power management logic performs Felter's modified pro rata technique. *Supra* §VII.A. This combination satisfies claim 12 because the power management logic, as modified, "throttles" (does not grant) requests to perform processor core activity that would "exceed" its activity "limit" during a first interval. *See* Felter, [0065], FIG. 3. After denying the processor core's request to perform more instructions during that time period

(denying a request to perform at a higher frequency), a POSITA would have at least found it obvious to increase that processor core's frequency during a subsequent interval by allocating more power to it. *See* Felter, [0051] (explaining "that very recent activity is a good predictor of current activity"), [0066] ("When the current interval expires, power administrator 200 generates new predictions for the next upcoming period … and the process repeats itself indefinitely."); Annavaram ¶¶177-182.

As explained for [8c], the Finkelstein-Felter combination allocates power to first and second power planes (domains) by accounting for "standby power requirements [$E^{k,i}{}_{\min}$] (which must always be met) and a portion of the available power [$E^k - \sum_i E^{k,i}_{min}$]." Felter, [0057]-[0061]; *see supra* §VII.B.1.d. As modified by Felter, the power management logic then "deriv[es] … activity limits that correspond to the power budgets allocated." Felter, [0064]; Annavaram ¶178. If any of the power planes "exceed" their limit, the power management logic would throttle activity "until the next monitoring interval begins." Felter, [0064]-[0066]; *see* FIG. 3; Annavaram ¶179.

Because "very recent activity is a good predictor of current activity," Felter discloses that "the predicted activity for time period N+1 [may be] a simple function of the monitored activity for time period N." Felter, [0051]. This facilitates "responsiveness in the context of a rapidly changing operating environment." *Id.*

Finkelstein discloses the same approach. *See* Finkelstein, [0028]-[0029] (formula (1)); Annavaram ¶180. Using recent activity to predict current utilization, a POSITA would have understood that if one of the power planes was throttled during the previous time interval because it lacked sufficient power, then its $W_{k,i}$ value should be incremented for the next interval to permit increased activity. *See* Annavaram ¶180 (referencing Ground 1, claim 12, and Ground 3, [8d]); Finkelstein, [0016] (explaining power management logic 140 monitors "operating frequency, operating voltage, power consumption, inter-core communication activity, etc.[] and/or information from one or more power monitoring logics 145" to "instruct the VR 130, power source 120, and/or individual components of system 100 (such as the cores 106) to modify their operations"). That is, if one of Finkelstein's power planes (domains) was prevented from performing more operations during the previous time interval (denying its request to perform at a higher frequency), then a POSITA would have understood it should be granted a higher frequency during the next interval. Annavaram ¶180.

Incrementing was a known, suitable option to increase frequency. As Dr. Annavaram explains, it was well known to increase the power budget of a domain when its power needs are not satisfied. Annavaram ¶181 (citing Therien, [0031] (describing a "software policy manager" that "periodically evaluates" a system "to determine whether any power budget adjustments are appropriate" and "make[s]

budget adjustments as conditions associated with the system 400 change (e.g., when a component raises or lowers an increase request flag)"); Conroy, [0006] ("[T]he power of the system may be shifted toward the subsystem having a higher workload. Thus, higher power draw is allocated to and allowed for the subsystem having a higher workload.")).

Thus, in the Finkelstein-Felter combination, a POSITA would have at least found it obvious to increment the first and second sharing policy values ($W_{k,i}$) when respective requests for a higher frequency for the first and second domains (first and second power planes, i=1 and i=2) are not granted (throttled). Annavaram ¶182.

3. Independent Claim 13

a. [13pre] (system)

The Finkelstein-Felter combination satisfies [13pre] because, as explained for Ground 2, Finkelstein discloses a "computing system" like the one below. Finkelstein, [0010]-[0014], [0042]-[0047], FIGS. 1, 6; Annavaram ¶¶183-184.

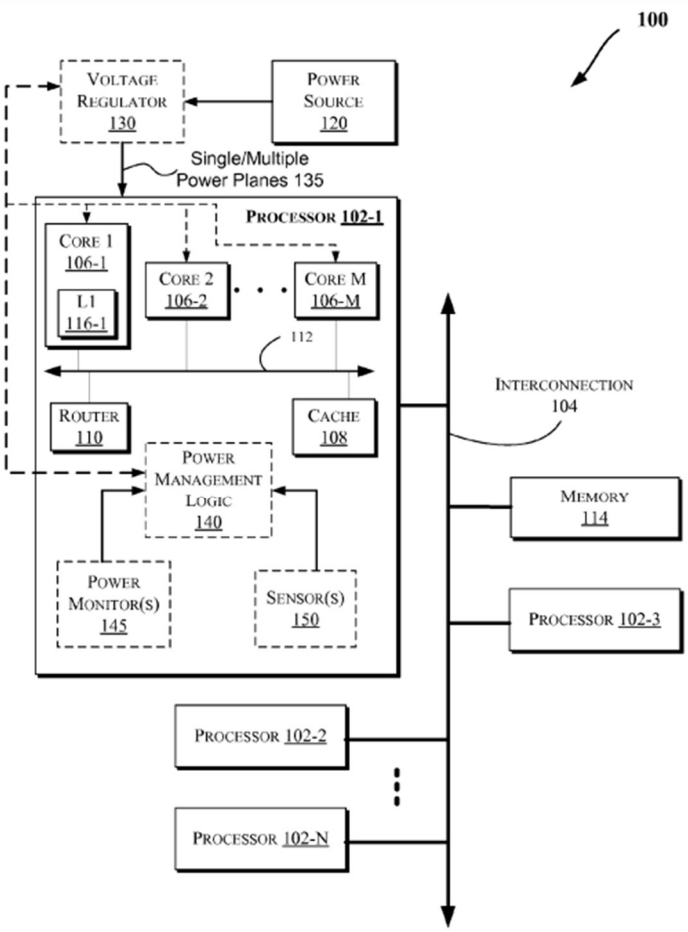


FIG. 1

Finkelstein, FIG. 1.

The Finkelstein-Felter combination employs Finkelstein's computing system, adapted so that its power management logic performs Felter's modified pro rata power allocation technique. *See supra* §VII.A.

b. [13a] (multicore processor with a "first" multicore domain, a "second" graphics engine domain, and a "third" system agent domain)

The Finkelstein-Felter combination satisfies [13a] because, as explained above for Ground 2, Finkelstein's computing system comprises a multicore, "multiple power plane" processor 102 including (1) a first power plane (domain)

having first and second cores 106-1 and 106-2, (2) a second power plane (domain) including a graphics controller for "Graphics Effect(s) (GFX) performance," and (3) a third domain including "power management logic" (system agent circuitry). Finkelstein, [0011], [0014], [0026], [0046], FIGS. 1, 6; Annavaram ¶185.

c. [13b] (system agent domain operates on a fixed power budget to allocate power to the multicore and graphics engine domains)

The Finkelstein-Felter combination satisfies [13b]. Annavaram ¶¶186-189. Finkelstein's third domain (Finkelstein's power management logic, as modified by Felter) dynamically allocates a variable power budget between a group of processor cores (first domain) and a graphics controller (second domain). Finkelstein, [0009], [0026], [0028], [0034]. Annavaram ¶187. The allocated power/energy budget is variable because it is determined "iteratively" for each new time interval based on power not consumed during the previous one. Finkelstein, [0028]-[0029]; *see* [0039]-[0040], FIG. 5; Annavaram ¶187.

Finkelstein's power management logic, as modified by Felter, receives information about various constraints (k), and it uses those constraints to determine a power/energy budget (E^k)[8] per constraint, which may be a package budget shared

[8] Although Finkelstein allocates an energy budget and Felter allocates a power budget, the two are equivalent. *Supra* notes 5-6; Annavaram ¶51, ¶187 n.6.

by the processor and graphics power planes. Finkelstein, [0026], [0034], [0039]; *see* Annavaram ¶188. As explained above, Finkelstein's power management logic, as modified by Felter, dynamically allocates the power/energy budget between the first and second domains (i=1, i=2):

$$\boldsymbol{E^{k,i} = E^{k,i}_{min} + W_{k,i}\left(E^{k} - \sum_i E^{k,i}_{min}\right)} \qquad \textbf{(Eq. 2, modified)}$$

See supra §VII.A; Annavaram ¶188. Accordingly, the third domain (power management logic) dynamically allocates a variable power budget (E^k) between the first and second domains (power planes). Annavaram ¶188.

For the reasons explained regarding [13b] in Ground 2, a POSITA would have at least found it obvious that Finkelstein's power management logic operates on a fixed power budget, which was routine in the art. *See supra* §VI.B.1.c (Ground 2, [13b]); Finkelstein, [0016], [0039]; Annavaram ¶189.

d. [13c] (system agent determines power budget for a current time interval)

The Finkelstein-Felter combination satisfies [13c] for the same reasons explained regarding [8a]. *See supra* §VII.B.1.b (Ground 3, [8a]). In terms of mapping the Finkelstein-Felter combination to claim 13, the only difference is that the variable power budget is shared by a processor power plane and a graphics controller power plane, not just any two processor power planes. *See supra* §VII.B.3.b-c; Finkelstein, [0046] (explaining the processor cores and graphics

controller may "share a common package power/energy budget"). But the budget is determined the same way. Annavaram ¶¶190-191.

Accordingly, the Finkelstein-Felter combination includes a power sharing logic (Finkelstein's power management logic, as modified by Felter) for determining a variable power budget (E^{k}_{n+1}) for a current time interval (n+1). *See* Finkelstein, [0028]-[0029], [0039]-[0040]; Annavaram ¶191.

e. [13d] (system agent allocates power to the multicore domain and graphics engine domain according to power sharing values)

As explained for [8c] and [8d], the Finkelstein-Felter combination allocates first and second portions ($E_{k,i}$) of the power budget (E_k) to the first and second domains (first and second power planes, i=1, i=2) according to first and second sharing policy values / power sharing values ($W_{k,i}$) and first and second minimum reservation values ($E^{k,i}_{min}$). *See supra* §VII.B.1.d-e (Ground 3, [8c] and [8d]); Annavaram ¶192.

f. [13e] (DRAM coupled to the multicore processor)

The Finkelstein-Felter satisfies [13e] because, as explained for [13e] in Ground 2, Finkelstein's computer system includes a "memory," which may be "dynamic RAM (DRAM)," that "communicate[s] with the processors 102 via [an] interconnection." Finkelstein, [0013], [0043], FIGS. 1, 6; Annavaram ¶193.

4. Claim 14 (substantially all the power budget is allocated to the multicore domain for a first workload and then to the graphics engine domain for a second workload)

The Finkelstein-Felter combination satisfies claim 14 because Finkelstein's power management logic, as modified by Felter, distributes the variable power budget in a "temporal" manner that accounts for each domain's workload for a prescribed time interval, including by allocating the "entire budget" to a "single power plane" when needed. Finkelstein, [0026]-[0028], [0034]; Annavaram ¶¶194-203.

The modified power management logic "continuously" or "periodic[ally]" determines a portion of the power budget to allocate to each power plane for a given "time period." Finkelstein, [0025], [0039]-[0040]; Annavaram ¶195. That is, it allocates the power budget dynamically "among [the] different power planes" from one time to the next. Finkelstein, [0027]. Thus, a POSITA would have understood the modified power management logic dynamically allocates the variable power budget (A) to the first and second domains based on a first workload that occurs during a first time period and (B) to the first and second domains based on a second workload executed after the first load (during a subsequent time period). *See* Annavaram ¶195; Finkelstein, [0028]-[0029], [0040]; Felter, [0066].

Finkelstein at least suggests allocating substantially all the variable power budget to the first domain for a first workload and then to the second domain for a

second workload. Finkelstein teaches to share a "***common package power/energy budget***" between processing and graphics power planes and explains "power management may need to be satisfied for … ***per package***." Finkelstein, [0026] (emphases added). According to Finkelstein, "the entire" budget may be allocated to one of the power planes during a time interval when appropriate. Finkelstein, [0034]; Annavaram ¶196. For example, "a ***single power plane*** may obtain ***the entire budget***" based on "user (or application defined) preferences that describe how budget E^k is distributed." Finkelstein, [0034]. Finkelstein thus at least suggests allocating substantially all the power budget to one power plane (e.g., processor cores) when the "user (or application defined) preferences" W_{ki} require as much for the first workload. Annavaram ¶196.

A POSITA would have understood that, for a processor-heavy workload, the expected activity (UTIL) for the processor cores would be high, whereas the expected activity for the graphics controller would be low. *See* Felter, [0059]-[0061], FIG. 5; Annavaram ¶197. Inversely, Finkelstein explains "unused processor core power" may be redistributed to the graphics controller "for more Graphics Effect(s) (GFX) performance." Finkelstein [0026]. That is, a POSITA would have understood the expected activity (UTIL) for the processor cores (first domain) would be low for a graphics-heavy workload, whereas the expected activity for the graphics controller would be high. *See* Felter, [0059]-[0061], FIG. 5; Annavaram ¶197. Accordingly,

the power sharing value for the first domain ($W_{k,1}$) would be near 0, whereas the power sharing value for the second domain ($W_{k,2}$) would be near 1, and substantially all the power budget would be allocated to the graphics power plane (second domain).

This was well known. Annavaram ¶198 (citing Gwennap, 6 ("If the GPU [(Graphics Processing Unit)] is not in use or is operating at a low speed, ***the power manager can assign some of its thermal budget to the CPUs***, allowing them to run at a higher clock speed. Conversely, ***on a graphics-intensive application, the power manager can overclock the GPU*** while slowing down the CPUs." (emphases added)). For example, a POSITA would have understood that many applications, like program development tools and scientific applications, produce a heavy load for processors and almost no load for a graphics controller. *See* Annavaram ¶199 (citing Conroy [0196]). Other applications, like advanced user interfaces or graphics editing applications, present the opposite situation. *See id*. (citing Conroy [0196]). Accordingly, "[m]any professional applications present an alternating high workload to the CPU and to the GPU." Conroy [0196].

Knowing this, a POSITA would have expected a multi-domain processing system to alternate between executing processor-heavy applications and graphics-heavy applications, allocating substantially all the power budget to one or more processor cores (first domain) for processor-heavy applications and substantially all

the power budget to the graphics controller (second domain) for graphics-heavy applications. *See* Finkelstein, [0026], [0034]; Felter, [0059]-[0061]; Annavaram ¶200.

To the extent Patent Owner disputes this or the Board determines otherwise, a POSITA would have been motivated to do so to ensure optimal system performance. *See* Annavaram ¶201 (citing Gwennap, 6; Conroy [0196]). Moreover, a POSITA would have reasonably expected to succeed because it only would have required routine skill. Annavaram ¶202 (citing Therien, [0014]; Conroy, [0191], [0195]-[0196]). By setting the power distribution preferences (W_{ki}) accordingly, *see* Finkelstein, [0034], a POSITA would have known how to configure the power management logic to allocate substantially all the available power (E^k_n) to the first domain for first (processor-heavy) workload, and subsequently to allocate substantially all the available power (E^k_{n+1}) to the second domain for a second (graphics-heavy) workload executed after the first (processor-heavy) workload. Annavaram ¶203.

5. Claim 15: "The system of claim 13, 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."

Claim 15 is similar to claim 12. As explained for claim 12, the power sharing values in the Finkelstein-Felter combination are calculated according to the expected

activity (UTIL) of each corresponding domain and would thus be incremented when the expected activity of the domain increased (to meet a request for a higher frequency). *See supra* §VII.A, §VII.B.2 (Ground 3, claim 12); Annavaram ¶¶177-182, ¶204.

6. Claim 16: "The system of claim 13, 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."

Claim 16 is similar to [8c]. The only difference in how the Finkelstein-Felter combination maps to claim 16 compared to [8c] is that the "second domain" includes a graphics controller rather than just any second processor core. *Compare supra* §VII.B.1.b (Ground 3, [8c]), *with supra* §VII.B.3.b (Ground 3, [13a]). This is immaterial. The Finkelstein-Felter combination renders obvious claim 16 because, as explained above regarding [8c] and [8d], its power management logic allocates first and second portions ($E_{k,i}$) of the power budget (E^k) to the first and second domains (first and second power planes, i=1 and i=2) according to first and second sharing policy values / power sharing values ($W_{k,i}$) and according to first and second minimum reservation values ($E^{k,i}{}_{\min}$). *See supra* §VII.B.1.d-e (Ground 3, [8c] and [8d]); Annavaram ¶205.

7. Claim 17 (further allocate power according to a preference value favoring the graphics engine domain over the multicore domain)

The Finkelstein-Felter combination at least suggests claim 17 because Finkelstein discloses prioritizing power distribution to the "GFX intensive workload, when processor resources are not fully utilized" and allocating power to a "single power plane" when appropriate or as "user (or application defined) ***preferences***" dictate. *See supra* §VI.B.3; Finkelstein, [0026], [0034] (emphasis added); Annavaram ¶¶206-207.

As explained for Ground 2, a POSITA would have known to program Finkelstein's power management logic to favor the second domain over the first domain (e.g., setting $W_{k,2}>W_{k,1}$). *See supra* §VI.B.3 (Ground 2, claim 17); Annavaram ¶207. A POSITA likewise would have found it obvious to configure the power management logic (as modified in the Finkelstein-Felter combination) to further allocate the available variable power budget ($E^k - \sum_i E_{min}^{k,i}$) to the second domain (graphics controller) according to a preference value (e.g., setting $W_{k,2}>W_{k,1}$, such that $W_{k,2}$ represents a preference value). *See supra* §§VII.B.1.d-e (Ground 3, [8c] and [8d]), §VI.B.3 (Ground 2, claim 17); Finkelstein, [0026], [0034]; Felter, [0051], [0059]-[0061], FIG. 5; Annavaram ¶207.

VIII. Ground 4: The Finkelstein-Felter-Therien Combination Renders Obvious Claims 12, 15, and 17

A. Overview of Therien (Ex-1007)

Therien, published on June 24, 2010, qualifies as prior art to the '316 patent under §102(b). It teaches methods for use in multicomponent (domain) computer systems. Therien, Title, Abstract, [0008]-[0010]. As in Finkelstein and Felter, such systems include "a first component 110," "a second component 120," and "a power budget allocation engine 150" to "facilitate an allocation of an overall power budget among the first and second components 110, 120 as appropriate." Therien, [0008]-[0010], FIG. 1.

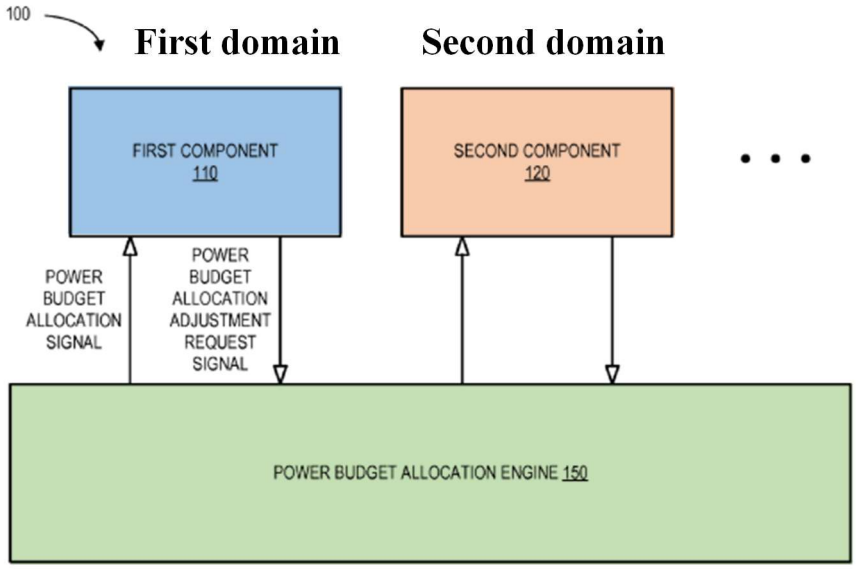


FIG. 1

Therien, FIG. 1 (annotated).

Therien's "power budget allocation engine" includes a "policy manager" that "manages a power and/or thermal budget associated with multiple components …

to facilitate improved results given a system preference." Therien, [0010], [0017], [0026], [0032]-[0034], FIGS. 4-5. To account for increased power demands, the processor and graphics components, for example, may send a "power budget allocation adjustment request signal," such as an "increase request flag," to the power budget allocation engine. Therien, [0011], [0015]. The power budget allocation engine may then "adjust a power budget allocation signal provided to the … component" so "an individual power limit imposed on a component … may be relaxed or raised as appropriate." Therien, [0013]-[0016], [0026], FIGS. 2, 4.

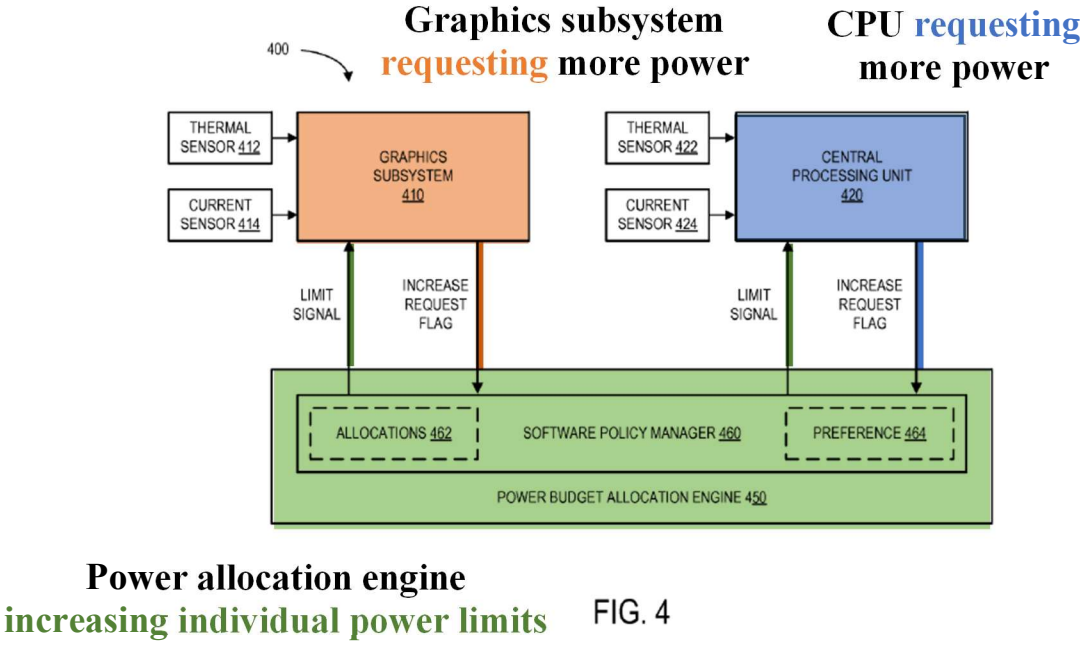


FIG. 4

Therien, FIG. 4 (annotated).

Therien is analogous art because it is in the same field of endeavor as the '316 patent (power allocation for microprocessors) and reasonably pertinent to a problem

it addresses (ensuring sufficient power). *Compare* Therien, Abstract, [0004], [0007], [0014]-[0017], *with* Ex-1001, Abstract, 1:18-22; *see* Annavaram ¶70.

B. Claims

1. Claim 15 (power sharing values are incremented when requests for higher frequencies are not granted)

To the extent Patent Owner argues Felter and/or Finkelstein do not sufficiently disclose or render obvious incrementing the power sharing values after a request for a higher frequency is denied, the Finkelstein-Felter combination, further combined with Therien, renders claim 15 obvious. A POSITA would have found it obvious from Therien to program the power management logic in the Finkelstein-Felter combination to employ such requests and to behave as claimed. Annavaram ¶¶208-212.

Like Finkelstein, Therien teaches power management methods for use in a system with a "graphics subsystem" and "CPU" (processor). *See* Therien, [0015], [0025]-[0026], FIGS. 4-5. Therien explains that adjusting the power budget allocation for a component may be appropriate if "an additional allocation of power could be usefully consumed." Therien, [0019]-[0020]. To request such adjustment, "an increase request flag can be raised," and the system may be granted an increase in frame rate or number of instructions retired per second (increased frequency) as a result. Therien, [0021]-[0022]; Annavaram ¶209. For example, Therien's power budget allocation engine may "adjust a power budget allocation signal provided to

the … component" so that "an individual ***power limit imposed on a component*** … may be relaxed or ***raised*** as appropriate." Therien, [0013]-[0016] (emphasis added). Annotated Figure 4 below illustrates this process. *See* Therien, [0011], [0015]-[0031], FIGS. 2, 4.

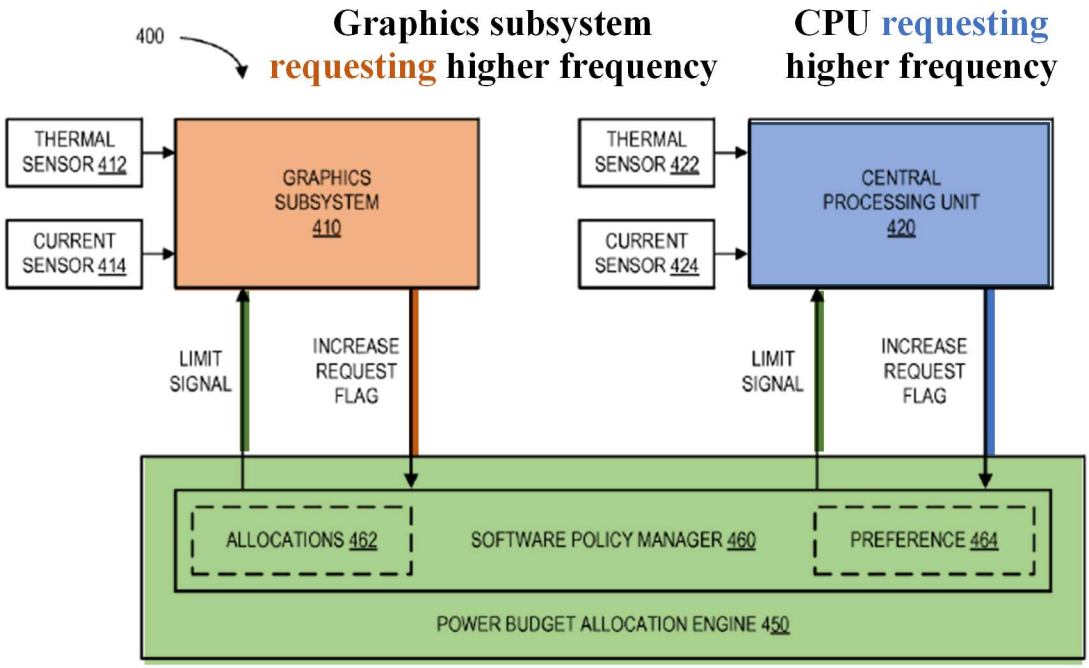


FIG. 4

Therien, FIG. 4 (annotated).

As explained for Ground 3, Finkelstein's power management logic, as modified by Felter, dynamically allocates power to various power planes according to their expected activity. *See supra* §VII.A; Felter [0051], [0057]-[0061]. As Felter teaches, recent activity is a good predictor of future activity. Felter, [0051]; *see* Finkelstein, [0028]-[0029]. If the power allocated to a plane was insufficient during

the previous interval, a POSITA would have understood "an additional allocation of power could be usefully consumed." Therien, [0019]. Accordingly, a POSITA would have understood the component should request additional power for the upcoming interval, as Therien teaches. *See* Therien, [0019]-[0031], FIGS. 3-4; Annavaram ¶210. A POSITA would have been motivated to do so by incrementing the component's predicted future activity (UTIL), thereby incrementing its sharing value ($W_{k,i}$). *See supra* §VII.A (describing the relationship between UTIL and $W_{k,i}$ in the Finkelstein-Felter combination); Felter, [0051], [0059]-[0061], [0066], FIG. 5; Annavaram ¶210.

A POSITA would have reasonably expected to succeed because it would have required only routine skill. Annavaram ¶211. As explained for Ground 3, Finkelstein's power management logic, as modified by Felter, already adjusts sharing policy values (or power sharing values) based on expected power plane activity (UTIL). *See supra* §VII.A; Felter [0051]-[0054], [0059]-[0061]. Incrementing the power sharing value to allocate additional power, e.g., to meet a request for a higher frequency (as Therien teaches) would have involved only known techniques (e.g., writing "lower level code, such as microcode") in accordance with a known use (power budget management). Therien, [0014]; *see* Finkelstein, [0039]; Felter [0051]-[0066], FIGS. 3, 5; Annavaram ¶211.

Accordingly, a POSITA would have found it obvious to configure the power management logic in the Finkelstein-Felter combination to increment the first and second power sharing values ($W_{k,i}$) when respective requests for a higher frequency for the first and second domains (processor power planes, i=1, i=2) are not granted. Annavaram ¶212.

2. Claim 12 (sharing policy values are incremented when requests for higher frequencies are not granted)

Claims 12 and 15 are similar. For essentially the same reasons explained in Section VIII.B.1 (Ground 4, claim 15), the combination of Finkelstein, Felter, and Therien renders claim 12 obvious because the sharing policy value ($W_{k,i}$) of a power plane (domain) i would be incremented if its request for a higher frequency was not granted. *See supra* §VIII.B.1; Finkelstein, [0034], [0039]; Felter, [0051], [0059]-[0061], [0066]; Therien, [0019]-[0022], FIG. 4; Annavaram ¶¶213-215.

A POSITA would have understood Therien's method is not limited to the case of a processor power domain and a graphics power domain; it applies equally to the case of two processor core domains. Annavaram ¶214. For example, Therien explains that "although some embodiments have been described with respect to particular components, any other types of components may be evaluated and/or allocated power." Therien, [0037]. Thus, although Therien describes embodiments comprising a processor domain and a graphics domain, it further discloses, or at least

suggests, other arrangements of components, including two processor core domains as the Finkelstein-Felter combination applies to claims 8 and 12. *Id*.

Further, even in Therien's embodiment with a processor domain and a graphics domain, a POSITA would have understood one of Finkelstein's processor cores can act as a graphics controller. *See* Annavaram ¶215 (citing Lalonde, [0023] ("[T]he use of GPU may be interchangeable with a CPU in some embodiments. For example, the processors 502 may include logic to support graphics functions."), FIG. 5 (same architecture as Finkelstein's FIG. 6)). Thus, for this additional reason, a POSITA would have understood Therien's teachings apply equally to the case of two processor domains as the Finkelstein-Felter applies to claims 8 and 12. *Id.*

3. Claim 17 (further allocate power according to a preference value favoring the graphics engine domain over the multicore domain)

To the extent Patent Owner argues Felter and/or Finkelstein do not sufficiently disclose or render obvious a "preference value," the Finkelstein-Felter combination, further combined with Therien, renders claim 17 obvious. Annavaram ¶¶216-220. A POSITA would have found it obvious from Therien to program the power management logic in the Finkelstein-Felter combination to employ a preference value as claimed. Annavaram ¶216.

As explained for Ground 2, Finkelstein prioritizes power distribution to a "GFX intensive workload, when processor resources are not fully utilized" and

allocates power to a "single power plane" when needed or as "user (or application defined) preferences" dictate. *See* Finkelstein, [0026]-[0028], [0034]. At least for graphics-heavy applications, this means $W_{k,2}>W_{k,1}$. *See* Finkelstein, [0034]; *supra* VI.B.3 (Ground 2, claim 17); Annavaram ¶217.

Therien expressly teaches that power management logic may "determine whether to adjust a power budget allocation … based [on] a ***pre-determined preference*** (e.g., ***indicating that graphics or computational performance should be given higher priority in the system*** 100)." Therien, [0012] (emphases added). Such determinations may be made "on a periodic basis … and/or upon a change in a power budget allocation adjustment request signal." Therien, [0013]. When the system is "configured such that a graphics subsystem will typically receive preferential power allocations as compared to a CPU," as Figure 5 shows below, the power budget allocation engine "may work together with various system components to facilitate an efficient allocation of power" based on the system preference. Therien, [0017]-[0018], [0032]-[0034], FIG. 5.

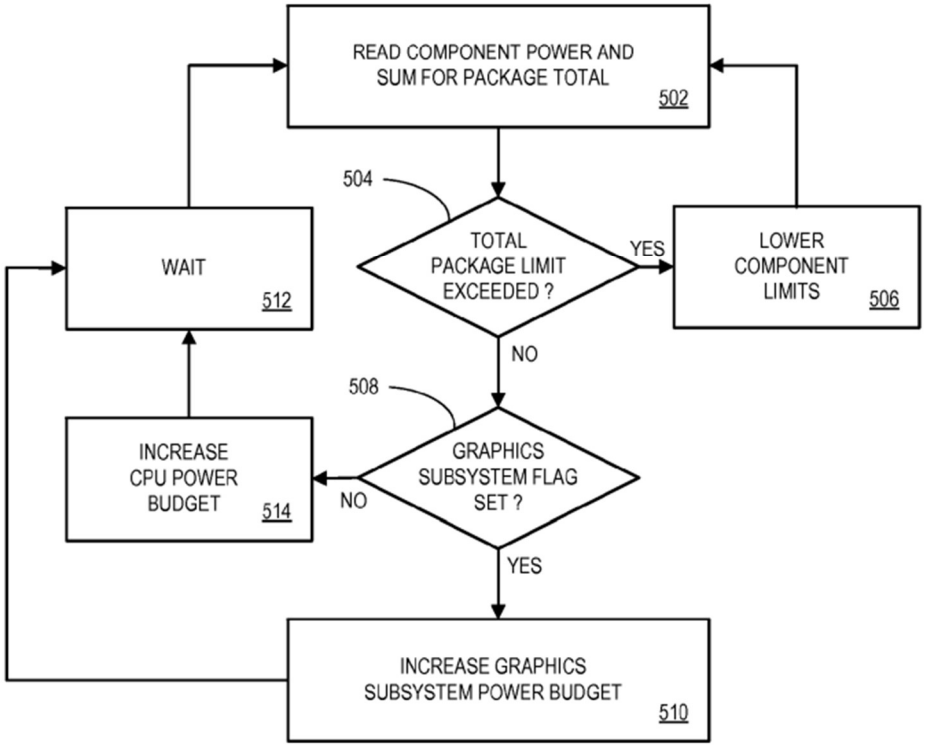


FIG. 5

Therien, FIG. 5.

A POSITA would have been motivated to allocate the available variable power budget according to a preference value that favors the graphics domain over the processor domain, as Therien teaches, "to facilitate an efficient allocation of power" and ensure the graphics controller has enough power to perform graphics-intensive functions. Therien, [0017]-[0018]; Annavaram ¶¶218-219. Such a preference value would have been particularly useful for graphics-heavy applications in the Finkelstein-Felter combination. *See* Finkelstein, [0026]; Annavaram ¶219.

A POSITA would have reasonably expected to succeed in configuring the power management logic to prioritize graphics operations because it only would have required routine skill to do so (e.g., setting $W_{k,2}>W_{k,1}$). *See supra* §VII.B.7 (Ground 3, claim 17); Annavaram ¶220. Indeed, Finkelstein envisions—and Therien confirms—that such preferences can be programmed based on user or application instructions. *See* Therien, [0017]; Finkelstein, [0034].

IX. Ground 5: The Felter-Therien Combination Renders Obvious Claim 12

A. Claims

1. Claim 12 (sharing policy values are incremented when requests for higher frequencies are not granted)

For the same reasons explained in Sections VIII.B.1 and VIII.B.2, the combination of Felter and Therien satisfies claim 12 because the sharing policy value ($PACT_N/PACT_{TOT}$) of a component (domain) would be incremented if its request for a higher frequency was not granted. *See supra* §VIII.B.1, §V.B.2; Felter, [0051], [0059]-[0061], [0066]; Therien, [0019]-[0022]; Annavaram ¶221. A POSITA would have found it obvious to incorporate Therien's teachings into Felter's power administrator for the same reasons it would have been obvious to incorporate them into the power management logic of the Felter-Finkelstein combination. Annavaram ¶¶208-215, ¶221; *see supra* §VIII.B.1, §VIII.B.2.

X. Ground 6: The Finkelstein-Therien Combination Renders Obvious Claims 15 and 17

A. Claims

1. Claim 15 (power sharing values are incremented when requests for higher frequencies are not granted)

For the reasons explained for Ground 4, the Finkelstein-Therien combination satisfies claim 15 because the sharing policy value (W_{ki}) of a power plane (domain) i would be incremented if its request for a higher frequency was not granted. *See supra* §VIII.B.1 (Ground 4, claim 15); Finkelstein, [0026], [0034]; Therien, [0019]-[0022], FIG. 3; Annavaram ¶¶222-224. The only difference compared to Ground 4, which is immaterial, is that a POSITA would have understood to increment Finkelstein's power sharing values (W_{ki}) according to Finkelstein's pro rata allocation technique rather than Felter's modified pro rata technique. *See supra* §VI.B.1.c (pro rata), §VII.A (modified pro rata); Finkelstein, [0034]; Annavaram ¶224.

2. Claim 17 (further allocate power according to a preference value favoring the graphics engine domain over the multicore domain)

For the reasons explained for Ground 4, the Finkelstein-Therien combination satisfies claim 17 because a POSITA would have found it obvious from Therien to program Finkelstein's power management logic to employ a preference value as claimed. *See supra* §VIII.B.3 (Ground 4, claim 17); Finkelstein, [0026], [0034]; Therien, [0017]-[0018]; Annavaram ¶¶225-226. The only difference compared to

Ground 4, which is immaterial, is that a POSITA would have understood to apply Finkelstein's pro rata allocation technique rather than Felter's modified pro rata technique. *See supra* §VI.B.1.c (pro rata), §VII.A (modified pro rata); Finkelstein, [0034]; Annavaram ¶226.

XI. Trial Should Be Instituted

A. Discretionary Denial Is Not Warranted Under *Fintiv*

Fintiv and the Office's interim guidelines dated June 21, 2022, favor institution.

Factors 1 and 5 are neutral. Petitioner is defendant in the parallel litigation. It remains to be seen whether the court would grant a stay if requested. The Board routinely declines to speculate whether a court would grant a stay.

Factor 2 does not disfavor institution. Although trial is tentatively scheduled to begin January 26, 2026, it is subject to change. Ex-1014, 1. The Board has instituted IPR on similar facts. *See Ericsson Inc. v. XR Commc'ns LLC*, IPR2024-00613, Paper 9, at 33-34 (P.T.A.B. Oct. 9, 2024) (trial scheduled approximately three months before FWD); *Samsung Elecs. Co. v. Staton Techiya, LLC*, IPR2022-01106, Paper 10, at 11 (P.T.A.B. Jan. 9, 2023) (same for trial scheduled two months before FWD); *Netnut Ltd. v. Bright Data Ltd.*, IPR2021-01492, Paper 12 at 9-16 (P.T.A.B. Mar. 21, 2022) (same for trial scheduled six months before FWD).

Factor 3 favors institution because Petitioner was diligent in filing this Petition, and the parties have made minimal investment in the parallel proceeding. Patent Owner filed its Complaint on April 8, 2024, and discovery only recently began on November 20, 2024. *See* Ex-1013, 163; Ex-1014, 4.

Factor 4 favors institution because the parties have not exchanged final contentions, and this Petition challenges nonasserted claims 13-17. Any overlap in issues at this stage is speculative.

Factor 6 favors institution because the merits of this petition are compelling.

Thus, discretionary denial is unwarranted under *Fintiv*.

B. Discretionary Denial Is Not Warranted Under *Advanced Bionics*

None of the references cited in this Petition were previously presented to or evaluated by the Examiner. Moreover, the asserted references show the Examiner erred in allowing the claims.

XII. Mandatory Notices

A. Real Parties-in-Interest Under 37 C.F.R. §42.8(b)(1)

MediaTek Inc. is the real party-in-interest for this Petition.

B. Related Matters Under 37 C.F.R. §42.8(b)(2)

To the best of Petitioner's knowledge, the '316 patent is or has been involved in the following case:

Name	Number	Court	Filed
Daedalus Prime LLC v. MediaTek, Inc.	2:24-cv-00235-JRG	E.D. Tex.	April 8, 2024

C. Lead and Backup Counsel Under 37 C.F.R. §42.8(b)(3)

Petitioner identifies the following lead and backup counsel:

Lead Counsel	Backup Counsel
Cory C. Bell (Reg. No. 75,096) cory.bell@finnegan.com Finnegan, Henderson, Farabow, Garrett, & Dunner LLP 2 Seaport Lane, 6th Floor Boston, MA 02210-2001 Tel.: 617-646-1641 Fax: 202-408-4400	J. Preston Long, Ph.D. (Reg. No. 65,125) jp.long@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 901 New York Avenue, NW Washington, DC 20001-4413 Tel: 202-408-4347 Fax: 202-408-4400 Safiya Aguilar (Reg. No. 77,212) safiya.aguilar@finnegan.com Finnegan, Henderson, Farabow, Garrett & Dunner, LLP 901 New York Avenue, NW Washington, DC 20001-4413 Tel: 202-408-4000 Fax: 202-408-4400

D. Service Information Under 37 C.F.R. §42.8(b)(4)

Please address correspondence to lead and backup counsel at the addresses above. Petitioner consents to electronic service by e-mail.

XIII. Standing

Petitioner certifies the '316 patent is available for *inter partes* review and Petitioner is not barred or estopped from requesting review on the grounds herein.

XIV. Conclusion

Trial should be instituted and the challenged claims cancelled.

Date: January 6, 2025

Respectfully submitted,

/Cory C. Bell/
Cory C. Bell
Lead Counsel
Reg. No. 75,096

CERTIFICATE OF COMPLIANCE

The undersigned hereby certifies that the foregoing **Petition for *Inter Partes* Review** contains 13,845 words, excluding those portions identified in 37 C.F.R. §42.24(a), as measured by the word-processing system used to prepare this paper.

/Cory C. Bell/
Cory C. Bell
Lead Counsel
Reg. No. 75,096

CERTIFICATE OF SERVICE

The undersigned certifies that the foregoing **Petition for *Inter Partes* Review,** associated **Exhibits 1001 through 1015**, and the **Power of Attorney** were caused to be served on January 6, 2025, by FedEx Priority Overnight, at the following address of record for the subject patent.

Attn: David Fairbairn
Kinney & Lange, P.A.
333 S. 7th Street, Suite 2700
Minneapolis, MN 55402-2438

Date: January 6, 2025

/Lisa C. Hines/
Lisa C. Hines
Case Manager
FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER, LLP