Appendix 316-04: Invalidity of U.S. Patent No. 8,769,316 (the ’316 Patent) Based on *Felter*

Appendix 316-04

The following list illustrates certain grounds for invalidity:

- U.S. Patent Application Publication No. 2006/0288241 to Felter et al. (“*Felter*”), filed on June 16, 2005, and published on December 21, 2006, qualifies as prior art to the ’316 patent under at least pre-AIA §§ 102(a), (b), and (e).

- U.S. Patent Application Publication No. 2010/0115304 to Finkelstein et al. (“*Finkelstein*”), which was filed on October 31, 2008, and published on May 6, 2010, qualifies as prior art to the ’316 patent under at least pre-AIA § 102(a), (b), and (e).

- U.S. Patent Application Publication No. 2011/0022356 to Nussbaum et al. (“*Nussbaum*”), which was filed on July 24, 2009, and published on January 27, 2011, qualifies as prior art to the ’316 patent under at least pre-AIA § 102(a) and (e).

- U.S. Patent Application Publication No. 2010/0064162 to Rotem et al. (“*Rotem*”), filed on September 5, 2008, and published on March 11, 2010, qualifies as prior art to the ’316 patent under at least pre-AIA §§ 102(a), (b), and (e).

- U.S. Patent Application Publication No. 2012/0023345 to Naffziger et al. (“*Naffziger*”), filed on July 21, 2010, and published on January 26, 2012, qualifies as prior art to the ’316 patent under at least pre-AIA § 102(e).

- U.S. Patent Application Publication No. 2012/0054511 to Brinks et al. (“*Brinks*”), which was filed on August 31, 2010, and published on March 1, 2012, qualifies as prior art to the ’316 patent under at least pre-AIA § 102(e).

- U.S. Patent Application Publication No. 2009/0089602 to Bose et al. (“*Bose-602*”), which was filed on September 27, 2007, and published on April 2, 2009, qualifies as prior art to the ’316 patent under at least pre-AIA §§ 102(a), (b), and (e).

As illustrated below, to the extent that *Felter* is found not to anticipate, expressly or inherently, one or more of the asserted claims of the ’316 Patent, each of claims 8-12 is invalid as obvious in view of *Felter* alone or in combination with other prior art references, including, but not limited to, the prior art discussed below, the prior art identified in Defendant’s Invalidity Contentions, and the prior art described in additional claim charts attached as Appendices 316-01a through 316-03 and 316-05 through 316-06. Exemplary invalidity grounds based on the *Felter* reference include the following, although these examples are by no means limiting:

- Obviousness based on *Felter* in combination with *Finkelstein* (claims 8, 9, 11, and 12);
- Obviousness based on *Felter* in combination with *Nussbaum* (claims 8, 9, 11, and 12);
- Obviousness based on *Felter* in combination with *Rotem* (claims 8, 9, 11, and 12);
- Obviousness based on *Felter* in combination with *Naffziger* (claims 8, 9, 11, and 12);
- Obviousness based on *Felter* in combination with *Finkelstein* and *Brinks* (claim 10);
- Obviousness based on *Felter* in combination with *Nussbaum* and *Brinks* (claim 10);
- Obviousness based on *Felter* in combination with *Rotem* and *Brinks* (claim 10);
- Obviousness based on *Felter* in combination with *Naffziger* and *Brinks* (claim 10);
- Obviousness based on *Felter* in combination with *Finkelstein* and *Bose-602* (claim 12);
- Obviousness based on *Felter* in combination with *Nussbaum* and *Bose-602* (claim 12);
- Obviousness based on *Felter* in combination with *Rotem* and *Bose-602* (claim 12); and
- Obviousness based on *Felter* in combination with *Naffziger* and *Bose-602* (claim 12).

The citations included in the table below are exemplary only and are not intended to be limiting or exhaustive. Defendant reserves the right to cite additional disclosures and evidence in support of their positions as this case develops further. Additionally, citation to a particular figure in a reference should be understood to encompass the caption and description of the figure and any text relating to or discussing the figure. Similarly, citation to any text referring to a figure should be understood to include the figure as well. Additionally, Defendant may be basing the arguments below on one or more claim construction positions that Daedalus has not expressly disclosed but seems to apply in its Patent Local Rule 3-1 disclosures or Defendant’s present understanding of Daedalus’s view and application of claim scope. By applying the prior art in this way, Defendant does not concede its agreement with Daedalus’s positions, nor do these

Invalidity Contentions constitute any admission regarding claim scope. Nor is Defendant asserting any claim construction positions through these charts. Moreover, claim preambles generally are not limiting, and nothing in these contentions should be deemed an admission that the preamble in the asserted claims is limiting. As this case further develops, Defendant may amend or supplement these exemplary disclosures and any related contentions as appropriate, and in a manner consistent with all applicable rules governing this case. In particular, Defendant reserves the right to rely on (1) uncited portions of the identified prior art; (2) other prior art not identified herein; (3) references that show the state of the art (irrespective of whether such references themselves qualify as prior art to the asserted patents); (4) factual testimony from the inventors or authors of the prior art references; and/or (5) expert testimony, to provide context to or aid in understanding the prior art and the state of the art at the time of the alleged inventions. Defendant also reserves the right to amend or supplement these exemplary disclosures based on changes in Daedalus’s positions as the case progresses, including changes to Daedalus’s infringement theories, claim interpretations, or apparent claim constructions.

Descriptions of Exemplary Obviousness Combinations Based on *Felter*

Felter and each of the additional references cited below are directed to device architectures with similar structures and functions. They are analogous art because they come from the same field of endeavor as the ’316 patent and are reasonably pertinent to the problems the ’316 patent addresses. Because they are all directed to the same types of device and have similar structures and functions, a person of ordinary skill in the art (“POSITA”) would have appreciated that *Felter* and the additional references cited below provide disclosures that relate to and complement one another. Moreover, a POSITA would have found it obvious to combine their disclosures in the ways discussed below.

Exemplary Reasons for Implementing *Felter*’s Power Management Method on a SOC

Felter teaches a “method for managing power in a data processing system having multiple components.” *Felter* at Abstract. *Felter* also states that the method is “applicable to single-chip systems in which multiple types of components (e.g., processor cores, cache memories, and the like) are integrated within a single piece of silicon.” *Felter* ¶ [0028]. Accordingly, a POSITA would have found it obvious to implement *Felter*’s power management method in a system-on-a-chip (“SOC”), such as the SOCs taught by, for example, *Finkelstein*, *Nussbaum*, *Rotem*, or *Naffziger*.

A POSITA would have understood that SOCs offer numerous advantages, such as compactness and productivity gains, from the ability to implement complex functions in a relatively short amount of time.[1] However, as the number of functionalities built into a single chip increases, so too does the heat generated by the system. *See Finkelstein* ¶ [0002]. Thus, a POSITA would have recognized the necessity of applying an efficient power management method, such as the one taught by *Felter*, to manage power in the system without exceeding thermal limits.[2] *Felter* explains that its method ensures “stable operation of the system within significantly lower power budgets and consequently smaller cooling resources while reducing the occasional negative impact of the reduced power budgets on system performance.” *Felter* ¶ [0009]. And, as already explained, *Felter*’s method is applicable to SOCs. *See Felter* ¶ [0028]. Thus, a POSITA would have expected to realize the benefits described by *Felter* when implementing *Felter*’s method on an SOC, such as the ones taught by, for example, *Finkelstein*, *Nussbaum*, *Rotem*, or *Naffziger*.

A person of ordinary skill in the art (“POSITA”) would have considered implementing *Felter*’s power management method on an SOC to be an example of (1) combining prior art elements (e.g., combining SOC hardware—as taught, for example, by *Finkelstein*, *Nussbaum*, *Rotem*, *Naffziger*, and others—with *Felter*’s power management method) according to known methods (e.g., implementing *Felter*’s power management method on an SOC) to yield predictable results (e.g., an SOC that ensures stable operation of the system within significantly lower power budgets and consequently smaller cooling resources); (2) a simple substitution of one known element (e.g., SOC hardware—as taught, for example, by *Finkelstein*, *Nussbaum*, *Rotem*, *Naffziger*, and others) for another (e.g., the data processing system hardware taught by *Felter*) to obtain predictable results (e.g., an SOC that ensures stable operation of the system within significantly lower power budgets and consequently smaller cooling resources); (3) use of a known technique (e.g., *Felter*’s power management method) to improve similar devices (e.g., SOC hardware—as taught, for example, by *Finkelstein*, *Nussbaum*, *Rotem*,

---

[1] *See Saleh*, et al., *System-on-Chip: Reuse and Integration*, Proceedings of the IEEE (Volume 94, Issue 6, June 2006) at 1050 (“Large productivity gains can be achieved using this SoC/IP approach. In fact, rather than implementing each of these components separately, the role of the SoC designer is to integrate them onto a chip to implement complex functions in a relatively short amount of time.”); *Martin*, et al., *System-on-Chip Design*, 2001 4th International Conference on ASIC Proceedings at 12 (“What motivates the move to SoC? The main motivation is the advance in silicon process technology that increasingly allows a complete system to be designed into one or a few integrated devices. Once this began to be possible, it was irresistible for designers and product managers to take advantage of the capability to map their products into highly integrated chips, taking advantage of space and power reductions, and increased performance.”).

[2] *See Finkelstein* at Abstract, ¶¶ [0002]-[0003], [0008]-[0009], [0016]-[0018], [0026]; *Nussbaum* at ¶¶ [0002]-[0004], [0017]-[0019], [0021]; *Rotem* at ¶¶ [0001]-[0002], [0008]-[0009], [0022]-[0023], [0027]-[0030]; *Naffziger* at Abstract, ¶¶ [0001]-[0007].

Naffziger, and others) in the same way (e.g., by implementing *Felter*'s power management method on an SOC to ensure stable operation of the system within significantly lower power budgets and consequently smaller cooling resources); (4) applying a known technique (e.g., *Felter*'s power management method) to a known device (e.g., SOC hardware—as taught, for example, by *Finkelstein*, *Nussbaum*, *Rotem*, *Naffziger*, and others) ready for improvement to yield predictable results (e.g., an SOC that ensures stable operation of the system within significantly lower power budgets and consequently smaller cooling resources); and (5) a teaching, suggestion, or motivation in the prior art (e.g., the exemplary teachings in *Finkelstein*, *Nussbaum*, *Rotem*, *Naffziger*, and others that SOCs should use an efficient power management method to manage power in the system without exceeding thermal limits) that would have led a person of ordinary skill in the art to modify *Felter* according to, or to combine *Felter* with, the teachings of *Finkelstein*, *Nussbaum*, *Rotem*, or *Naffziger*, for example, to arrive at the claimed invention. *See, e.g.*, *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 416-21 (2007).

Exemplary Reasons for Obtaining a Minimum Reservation Value from a Configuration Register Written by User-Level Software in *Finkelstein*'s Power Allocation Method

Under *Felter*, the minimum reservation value of domain N is its standby power (PMIN$_N$). *See infra* [8c]. *Felter* explains that the standby power is a minimum power value that must always be provided to domain N. *See Felter* ¶ [0058], ¶ [0061]. The standby power may include "a reservoir for minimal component activity . . . to ensure there is always enough power for each component to obtain some minimal work from it." *Felter* ¶ [0065]. A POSITA would have found it obvious to set this minimum reservation value by obtaining it from a configuration register written by user-level software.

For example, *Brinks*—like *Felter* and the '316 patent—teaches "an intelligent power controller for a complex electrical system, including System on a Chip, through power and clock distributions." *Brinks* ¶ [0002]. That is, a "power manager may . . . control power management entities that have a dynamic and controllable effect, including clock gating, power gating, dynamic voltage and dynamic frequency scaling." *Id.* ¶ [0022].[3]

Figure 4 below illustrates an exemplary hierarchy of *Brinks*'s power manager. *Brinks* ¶ [0047]; *see also id.* ¶ [0059], Fig. 5. It includes a policy manager 402, power management policies 404, and operating points 406. *Id.* ¶ 0047]; *see also id.* ¶ [0059].

---

[3] *see also* '316 pat. at 5:3-7 ("[A] percentage of the package budget . . . can be allocated to each of the domains by controlling their frequency and/or voltage accordingly."); *Nussbaum* ¶ [0033] ("A performance state is characterized by a unique pair of core voltage

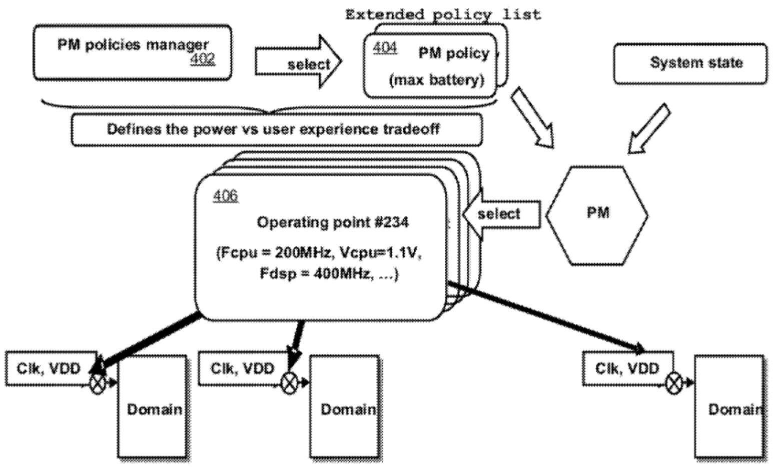


FIGURE 4

The power management policies “may be stored in a table, hierarchical table, register, matrix or some combination of the three that map different system operating states to different operating points 406 to improve and/or maximize energy savings for a given policy.” *Id.* ¶ [0049]; *see also de Cesare* ¶ [0040] (“In some embodiments, the registers 30B may directly store values defining the performance state to be established in the corresponding power domain.”). As one example, the power management policies “may be configured and/or programmed . . . by a chip designer” by “run[ning] multiple simulations to perfect knowing which components affect other components when the chip runs and how much power is saved.” *Brinks* ¶ [0049]. This is, for instance, how *Felter* determines PACT_{N} values.[4]

---

and frequency values.”), Tbl. 3; *Felter* ¶ [0037] (“Outlined below is an approach to using DVFS [dynamic voltage and frequency scaling] . . .”).

[4] *See Felter* ¶ [0054] (“[A]llocating of power budgets in block 306 utilizes a database (326) that includes information enabling one to convert activity levels for a specific component into a power consumption estimate for that component. Database 326 may be, for

To supplement such training performed by the chip designer, *Brinks* further teaches that “power management policies may be initially set after the simulations, but may also be updated during device operation based on actual user feedback or additions of new applications executing on the integrated circuit.” *Brinks* ¶ [0050]. That is, “the table, register, or matrix may be programmable, so that its behavior can be changed at run-time . . . and by a user during operation.” *Id.*; *see also id.* ¶ [0061]. Or, put another way, the power management policies may be updated using user-level software. *See id.* ¶¶ [0050], ¶ [0061]; *see also de Cesare* ¶ [0008] (“[A] power management unit may be configured to automatically transition (in hardware) the performance states of one or more performance domains in a system. The target performance states to which the performance domains are to transition may be programmable . . . by software.”).

As *Brinks* explains, power allocation variables may be updated “during device operation based on actual user feedback or additions of new applications executing on the integrated circuit.” *Brinks* ¶ [0050]. In other words, a POSITA would have found it obvious to provide user-level software control over *Felter*’s power allocation variables (e.g., power budget P_{BUDGET}, minimum reservation values $PMIN_N$, and/or power sharing values $PACT_N$). *See id.* ¶ [0051] (“[T]he chip designer or user will be responsible for providing the power management policies.”).

A POSITA would have been motivated to do so because the power management policies define “the tradeoffs between user experience and power consumption,” and that a user should be able to make decisions that “appropriately balance these tradeoffs.” *Brinks* ¶ [0051]. *Brinks* further teaches that power demand decisions may be “application dependant [sic],” *id.* ¶ [0053], and that user-level control over power allocation provides “a desired user experience, for instance, to maximize energy savings, or as another example, to maximize system reactivity,” *id.* ¶ [0048]; *see also de Cesare* ¶ [0009] (“[T]he configureability [sic] of the performance states may permit more fine-grained control of the performance level in the system and thus may permit additional power savings.”). A user may, for instance, prefer to allocate power in a way that “favor[s] faster performance over battery life, or vice-versa.” *Id*.

Further, a POSITA would have reasonably expected to succeed in providing such user-level software control over *Felter*’s power allocation variables, because those power allocation variables—namely, total power budget (P_{BUDGET}), dynamic sharing policy variables ($PACT_N$), and minimum “standby” power values ($PMIN_N$)—are flexible enough to allow such user-level software control.

---

example, an empirically derived lookup table or a more complex predictive model of power consumption as a function of activity. . . . [D]atabase 326 may reflect device specific characteristics that influence a device’s power consumption relative to other similar devices.”), ¶ [0059] (“[A]llocating the system power includes using the values of predicted activity . . . to compute the amount of active power per component expected during the upcoming interval.”).

For example, although the maximum possible system power budget may be set by the thermal limits of the system,[5] a POSITA would have found it obvious to set a lower P_{BUDGET} via user-level software if the user would like to conserve battery power, for instance.[6]

Similarly, a POSITA would have understood that certain power allocation decisions may be “application dependant [sic]” such that those decisions “may be under user control.” *Id.* ¶ [0053]; *see also Finkelstein* ¶ [0034] (explaining that user preferences may be used to describe how the power budget is distributed among power planes). For example, a POSITA would have understood that “a reservoir for minimal component activity can be included in the standby power allocated for the component [i.e., $PMIN_N$] to ensure there is always enough power for each component to obtain some minimal work from it.” *Felter* ¶ [0065]. Since the amount of power to reserve may depend on the application and/or a user’s preferences for balancing performance vs. battery usage, a POSITA would have found it obvious to allow user-level software control over such decisions. *See Brinks* ¶ [0051], ¶ [0053].

Brinks further teaches that power management policies “may be configured and/or programmed into [a] table, register, or matrix” that “map[s] different system operating states to different operating points.” *Id.* ¶ [0049]; *see also id.* ¶ [0040] (“The power manager may also include software related system states . . . communicated from software layers to the power manager hardware layers by register access.”), ¶ [0050] (explaining that “the table, register, or matrix may be programmable, so that its behavior can be changed at run-time” and that “multiple power management policies may be programmed and or updated . . . by a user during operation.”), ¶ [0071] (“[T]he operating points are named data structures expressing the intended power state of the domain given a particular system state and under a given power management policy stored in software visible registers, tables in memory, or any combination of both.”). Since, as explained above, the power allocation variables, including $PMIN_N$, relate to the power management policies, a POSITA would have found it obvious to write the minimum reservation value for the second domain into a configuration register via user-level software and for power allocation controller 109 to obtain the minimum reservation value for the second domain by reading that configuration register.[7]

---

[5] *See e.g., Felter* ¶ [0057]; *Finkelstein* ¶ [0031]; *Nussbaum* ¶ [0021], ¶ [0036]; *Rotem* ¶¶ [0008]-[0009], ¶¶ [0027]-[0028], ¶¶ [0032]-[0034] ¶ [0049]; *Naffziger* ¶ [0006], ¶ [0022], ¶ [0024], ¶ [0030], ¶ [0037].

[6] *See Brinks* ¶¶ [0036]-[0037], ¶¶ [0050]-[0051], ¶ [0077]; *Nussbaum* ¶ [0021] (“A TDP (Thermal Design Point) . . . depends on such factors as . . . AC adapter/battery . . . .”); *Finkelstein* ¶ [0029] (“[T]he TDP values may be provided by a software application or user in some embodiments.”).

[7] *See Brinks* ¶¶ [0040]-[0041], ¶¶ [0049]-[0051], ¶ [0061], ¶ [0071].

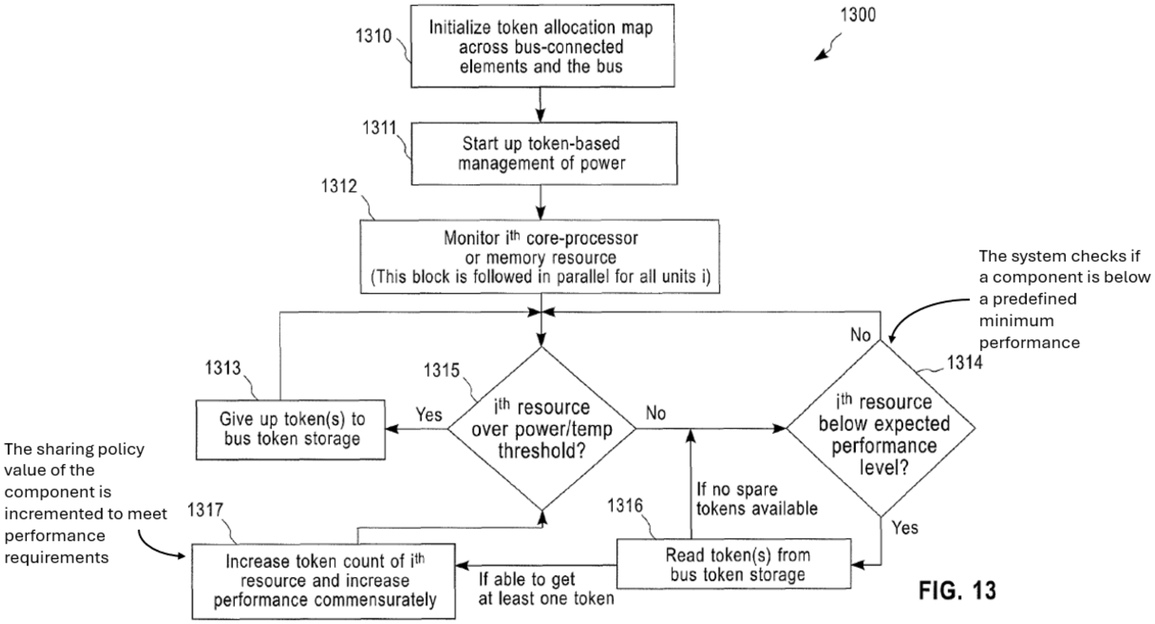
A person of ordinary skill in the art would have considered obtaining the minimum reservation value for the second domain from a configuration register written by user-level software to be an example of (1) combining prior art elements (e.g., combining *Felter*’s power allocation method, and providing user-level software control over power allocation in a multi-domain SoC as taught by, for example, *Brinks*) according to known methods (e.g., using software to write power allocation variables to corresponding configuration registers for use in power allocation determinations, as taught by, for example, *Brinks*) to yield predictable results (e.g., more fine-grained control of power allocation in a multi-domain SoC, allowing for a more satisfying user experience based on user preferences); (2) a simple substitution of one known element (e.g., *Brinks*’s technique of using user-level software and pre-defined models to determine power allocation variables in a multi-domain SoC) for another (e.g., the technique of relying on pre-defined models to determine power allocation variables in a multi-domain SoC, as used by *Felter*) to obtain predictable results (e.g., more fine-grained control of power allocation in a multi-domain SoC, allowing for a more satisfying user experience based on user preferences); (3) use of known technique (e.g., *Brinks*’s technique of using user-level software and pre-defined models to determine power allocation variables in a multi-domain SoC) to improve similar devices (e.g., a processor—as taught, for example, by *Finkelstein*, *Nussbaum*, *Rotem*, *Naffziger*, and others—implementing *Felter*’s power allocation method, which relies on pre-defined models to determine power allocation variables) in the same way (e.g., by allowing user-level software to help determine power control variables); (4) applying a known technique (e.g., *Brinks*’s technique of using user-level software and pre-defined models to determine power allocation variables in a multi-domain SoC) to a known device ready for improvement (e.g., a processor—as taught, for example, by *Finkelstein*, *Nussbaum*, *Rotem*, *Naffziger*, and others—implementing *Felter*’s power allocation method, which relies on pre-defined models to determine power allocation variables) to yield predictable results (e.g., more fine-grained control of power allocation in a multi-domain SoC, allowing for a more satisfying user experience based on user preferences); and (5) a teaching, suggestion, or motivation in the prior art (e.g., *Brinks*’s teaching that providing both user-level software control and pre-defined hardware control over power policy variables accounts for actual user feedback, improves user experience, and accounts for tradeoffs between performance and power conservation) that would have led a person of ordinary skill in the art to modify *Felter* according to, or to combine *Felter* with, the teachings of *Brinks*, for example, to arrive at the claimed invention. *See, e.g.*, *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 416-21 (2007).

Exemplary Reasons for Incrementing the Sharing Policy Value when a Request for a Higher Frequency is not Granted

Under *Felter*, the sharing policy value of domain N is its dynamic power (PACT_N). *See infra* [8d]. A POSITA would have found it obvious to increment the PACT_N value when a request for a higher frequency is not granted as taught, for example, by *Bose-602*.

Bose-602 teaches allocating a power budget to components by “initializing a token allocation map across the connected components, wherein each component is assigned a power budget as determined by a number of allocated tokens in the token allocation map.” *Bose-602* ¶ [0013]. *Bose-602*’s method monitors processor cores, assigns more tokens to the cores that can make use of more power, and then increases performance based on the number of tokens. *See id.* at Fig. 13.

According to *Bose-602*’s technique, the system checks whether a component is operating “below a predefined minimum performance.” *Id.* ¶ [0086]; *see also id.* at Fig. 13 (block 1314). If that component is operating below the predefined minimum performance, “extra tokens are used to boost up performance in the targeted unit . . . by increasing its voltage and frequency.” *Id.* ¶ [0086]; *see also id.* at Fig. 13 (block 1317). Once extra tokens have been assigned and performance has been increased, *Bose-602*’s method again checks whether the component is operating within the power and temperature limits and, if so, also whether it is operating below a predefined minimum performance. *See id.* ¶ [0086], Fig. 13 (blocks 1314 and 1315). If the component is still operating below the predefined performance minimum, more tokens are allocated to it. Accordingly, *Bose-602* teaches incrementing the number of tokens allocated to the component to meet a minimum performance requirement (which would include a request for a higher power and, thus, a higher frequency).



The system checks if a component is below a predefined minimum performance

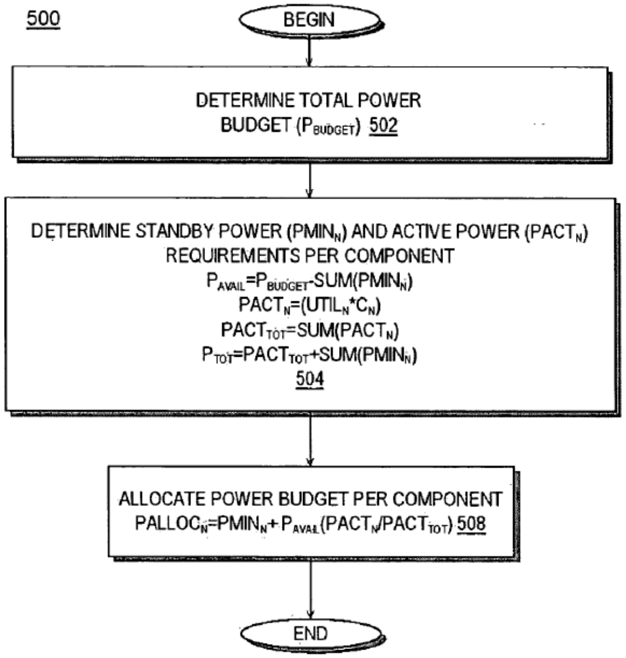
The sharing policy value of the component is incremented to meet performance requirements

FIG. 13

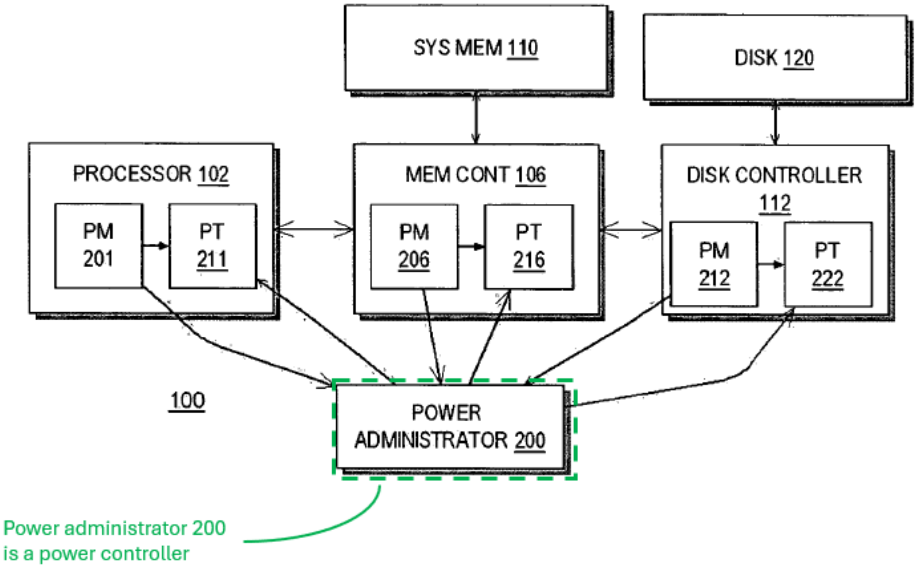
Like *Felter*’s sharing policy values (PACT$_N$), the number of *Bose-602*’s tokens determines the portion of the power budget allocated to a component. A POSITA would have found it obvious to borrow *Bose-602*’s teaching and increment *Felter*’s PACT$_N$ values.

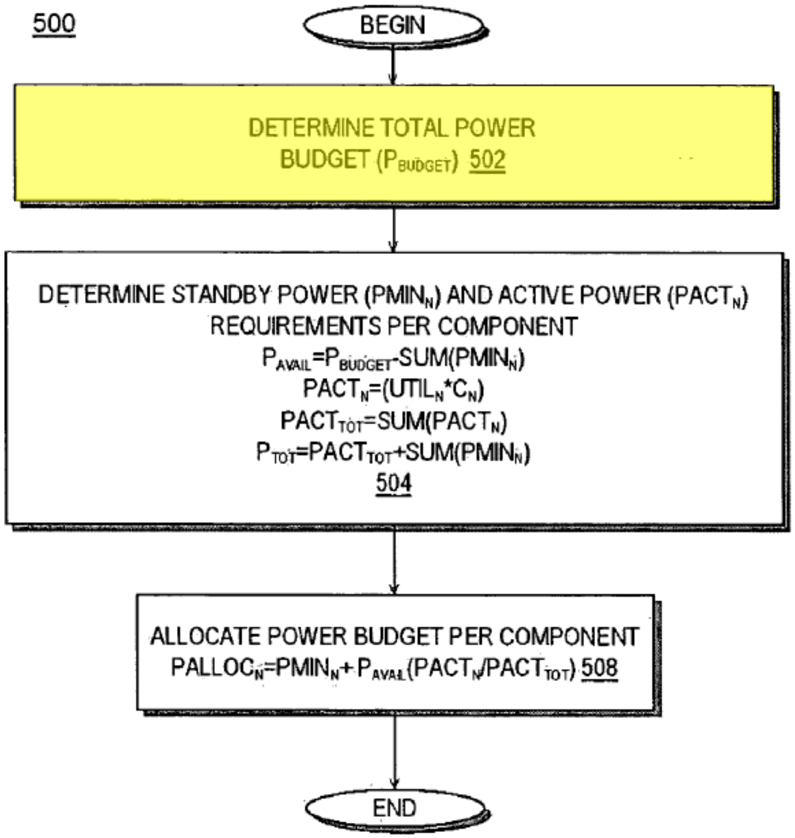
A POSITA would have been motivated to do so to ensure the domains necessary to process a given workload during a time interval can meet the relevant performance requirements. As *Felter* teaches, for example, PACT$_N$ refers to the “expected level of active power consumption” required by domain N. *Felter* ¶ [0060]. And the power controller allocates power based on “the ratio of the components active power requirements to the total active power requirements of the system.” *Id.* ¶ [0061]. If a domain requests a higher frequency, a POSITA would have understood that means the domain requires more power to meet performance requirements. *See e.g.*, *Felter* ¶¶ [0037]-[0039] (explaining that power is a function of voltage and frequency); *Nussbaum* ¶ [0037] (“[T]he power of a particular computational unit (voltage×current) is based on the frequency of the clock signal, the supply voltage, and the amount of activity in the computational unit.”). To meet those performance requirements, a POSITA would have found it obvious to apply *Bose-602*’s teachings—e.g., by incrementing the sharing policy values (PACT$_N$, including PACT$_1$ and PACT$_2$), thereby ensuring each domain (including the first domain and the second domain) can receive more power (a higher frequency) to dynamically satisfy increased performance demands.

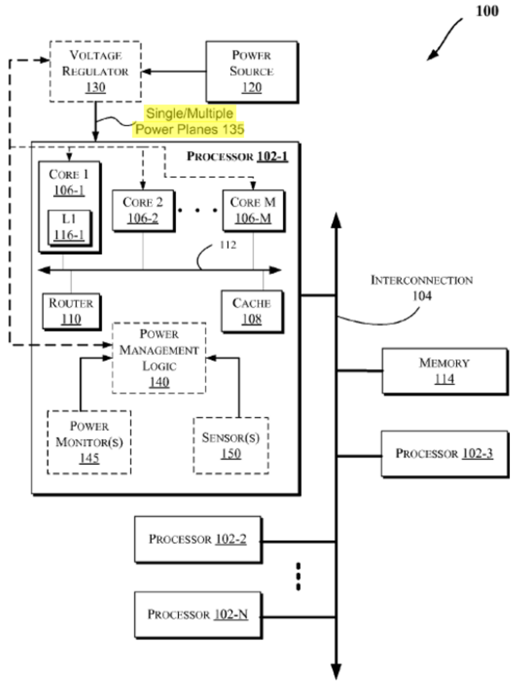
A POSITA would have considered incrementing *Felter*’s PACT$_N$ values when a request for a higher frequency was not granted to be an example of (1) combining prior art elements (e.g., *Felter*’s power management method, which allocates power based on PACT$_N$, with *Bose-602*’s teaching to increment the allocated power allocated when performance requirements are not met) according to known methods (e.g., checking whether each component meets performance requirements and, if not, incrementing the power allocated to the associated domain(s)) to yield predictable results (e.g., a multicore processor that satisfies the given workload demands over a given processing interval); (2) use of known technique (e.g., *Bose-602*’s teaching to increment the power allocated to a domain if a performance requirement is not met) to improve similar devices (e.g., a multi-core processor that implements *Felter*’s power management method) in the same way (e.g., incrementing the PACT$_N$ value when higher performance is required to meet a frequency request by one of the components, such that the request may be satisfied); (3) applying a known technique (e.g., *Bose-602*’s teaching to increment the power allocated to a domain if a performance requirement is not met) to a known device (e.g., a multi-core processor that implements *Felter*’s power management method) ready for improvement to yield predictable results (e.g., a multi-core processor that satisfies workload demands over a given processing interval); and (4) a teaching, suggestion, or motivation in the prior art (e.g., *Bose-602*’s teaching to increment the power allocated to a domain if a performance requirement is not met) that would have led a person of ordinary skill in the art to modify *Felter* according to, or to combine *Felter* with, the teachings of *Bose-602*, for example, to arrive at the claimed invention. *See, e.g.*, *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 416-21 (2007).

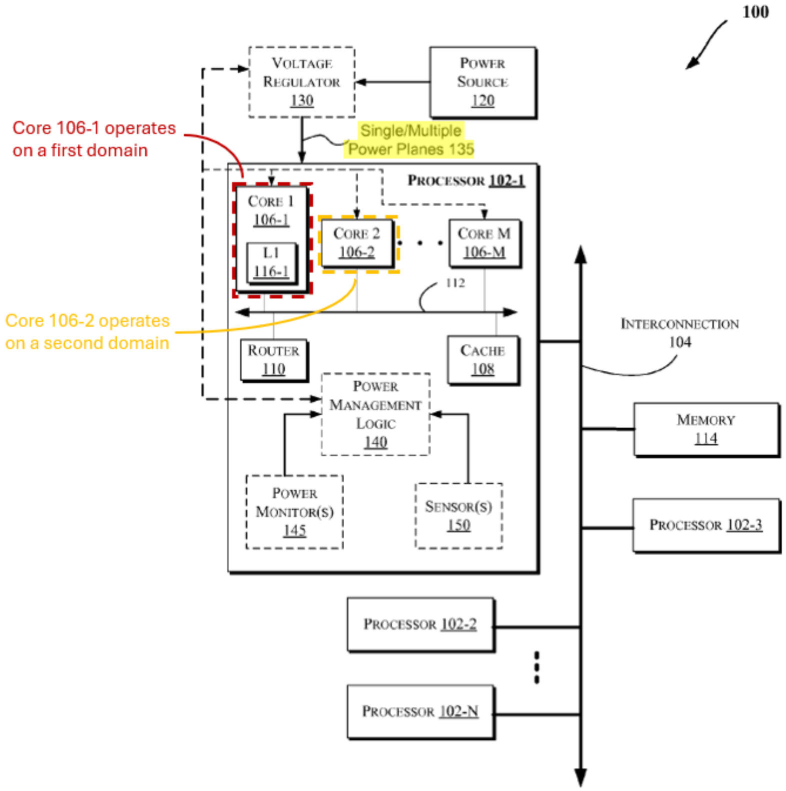
U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
Claim 8	
[8pre] A method comprising:	*Felter*, alone or in combination with one or more references, discloses or suggests a "method." For example, *Felter* describes a "method for managing power in a data processing system having multiple components." *Felter* at Abstract. Specifically, *Felter* teaches "a method for allocating power budgets to the system components based on expected levels of activity." *Felter* ¶ [0016], *see also Felter* ¶¶ [0057]-[0061], Fig. 5.  **Exemplary teachings from the prior art include the following:**

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	Felter discloses the following at Abstract: A method for managing power in a data processing system having multiple components includes determining a power budget for the system. Activity levels during a forthcoming time interval are then predicted for each of the components. Using the predicted activity levels, the power budget is allocated among the system components. An activity limit is then established for each component based on its corresponding portion of the power budget. The activity of a component is then monitored and, if the component's activity exceeds the component's corresponding activity limit, constrained. Determining the predicted level of activity may include determining a predicted number of instructions dispatched by a processor component or a predicted number of memory requests serviced for a system memory component. Allocating the power budget 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* discloses the following at ¶ [0016]: FIG. 5 is a flow diagram of a method for allocating power budgets to the system components based on expected levels of activity; and *See also Felter* at Abstract, ¶¶ [0057]-[0061].
[8a] determining, in a power controller of a multi-domain	*Felter*, alone or in combination with one or more references, discloses or suggests “determining, in a power controller of a multi-domain processor, a power budget for the multi-domain processor for a

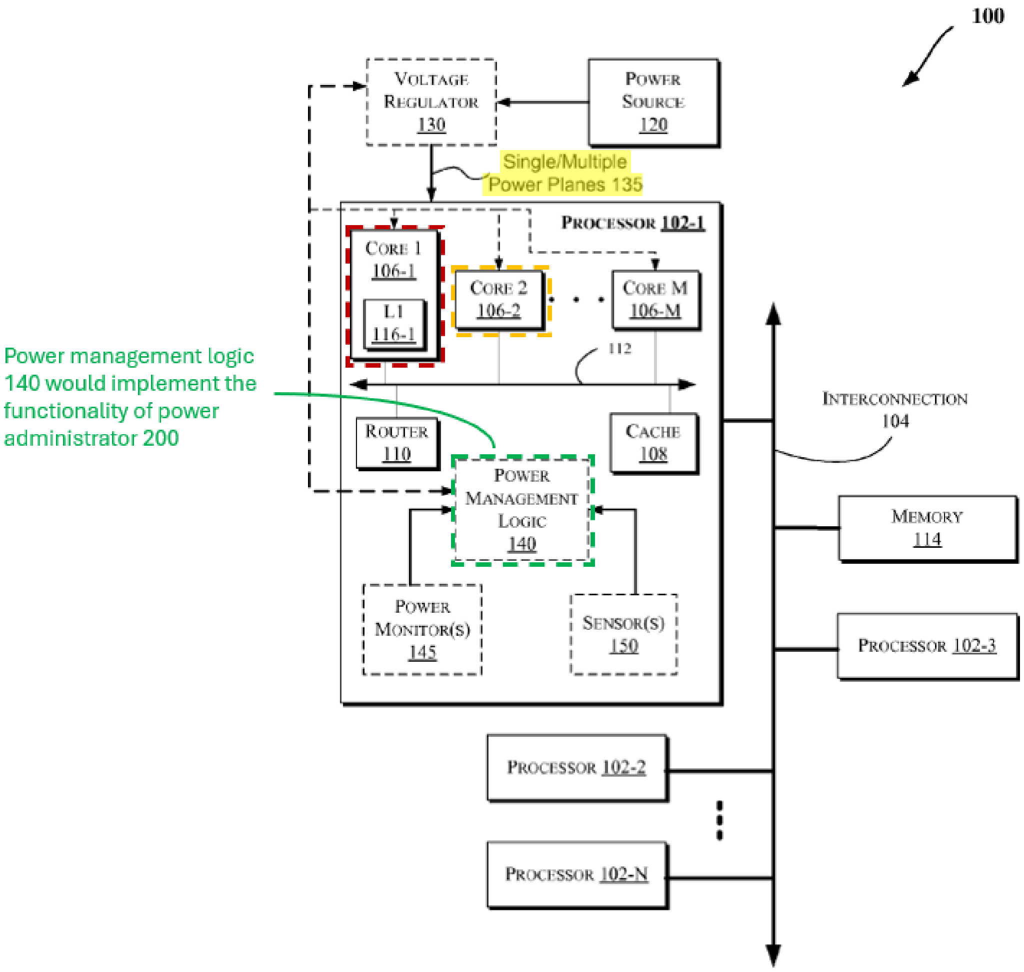
U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
processor, a power budget for the multi-domain processor for a current time interval, the multi-domain processor including at least a first domain and a second domain;	current time interval, the multi-domain processor including at least a first domain and a second domain." For example, *Felter* includes a power administrator 200. *See Felter* ¶ [0030], Fig. 2. Power administrator 200 "allocates the available power to system components based on predicted levels of activity for each component." *Felter* ¶ [0030]. "[P]ower administrator 200 may be implemented as an integrated or separate on-chip/on-board programmable or hardcoded microcontroller." *Id*. Thus, power administrator 200 is a power controller that controls the allocation of power to system components.  FIG. 2

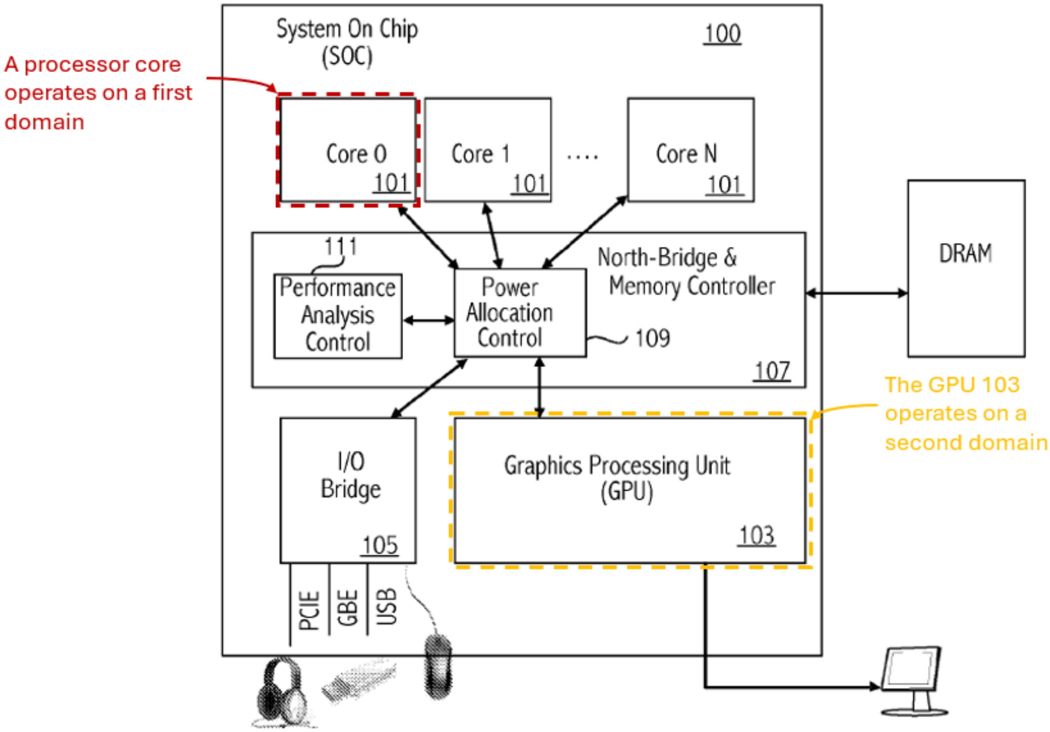
U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	Felter also teaches that its power allocation method includes determining a power budget. *See Felter* ¶ [0057] (“[M]ethod 500 includes determining (block 502) the total power budget (P_{BUDGET}) of the system.”), Fig. 5. 

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	Felter further explains that the “disclosed power management method is also applicable to single-chip systems in which multiple types of components (e.g., processor cores, cache memories, and the like) are integrated within a single piece of silicon.” *Felter* ¶ [0028]. As explained above, a POSITA would have found it obvious to implement *Felter*’s method on a multi-domain processor, such as the ones taught by *Finkelstein*, *Nussbaum*, *Rotem*, and *Naffziger*. *See supra* **Exemplary Reasons for Implementing *Felter*’s Power Management Method on a SOC**. For example, *Finkelstein* teaches a processor with multiple power planes 135:  FIG. 1

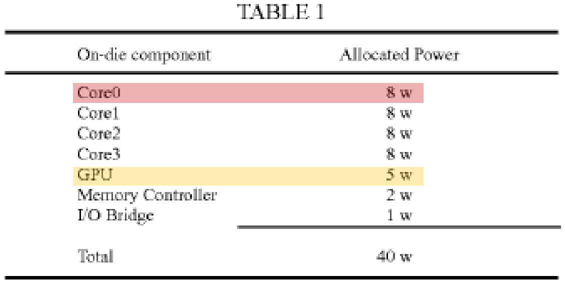
U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	Finkelstein teaches that “voltage regulator(s) 130 may be coupled to a single power plane 135 (e.g., supplying power to all the cores 106) or to multiple power planes 135 (e.g., where each power plane may supply power to a different core or group of cores).” *Finkelstein* ¶ [0014]. A POSITA would have understood that a power plane is a power domain. *See, e.g.*, ’316 pat. at 4:3-4 (explaining domains may “also be referred to as ‘planes’”).  FIG. 1

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	Finkelstein also teaches that each of the power planes operates at its own voltage and frequency. For example, *Finkelstein* teaches that logic 140 may “throttle or modify an operational characteristic of one or more of the cores 106 (such as an operating voltage and/or an operating frequency of a processor core 106).” *Finkelstein* ¶ [0024]. Because the voltage and frequency of one or more processor cores 106 may be controlled independently of others, a POSITA would have understood those processor cores 106 operate at a voltage and frequency controlled independently of other power planes. *See also, e.g.*, *Nussbaum* ¶ [0033] (“A performance state is characterized by a unique pair of core voltage and frequency values.”), Tbl. 3. Therefore, core 106-1 would operate on a first domain and core 106-2 would operate on a second domain. Further, *Finkelstein* describes a power management logic 140 that may “control supply of power to components of the processor 102 (e.g., cores 106).” *Finkelstein* ¶ [0016]; *see also id.* at Fig. 1. According to *Finkelstein*, “the operations discussed herein, e.g., with reference to FIGS. 1-7, may be implemented as hardware (e.g., logic circuitry),” thereby indicating power management logic 140 may be implemented as a “power controller,” as claimed. *Finkelstein* ¶ [0053]. Given their similarity in purpose, a POSITA would have understood that *Finkelstein*’s power management logic 140 would implement the functionality of *Felter*’s power administrator 200.

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	 Power management logic 140 would implement the functionality of power administrator 200 Single/Multiple Power Planes 135 FIG. 1

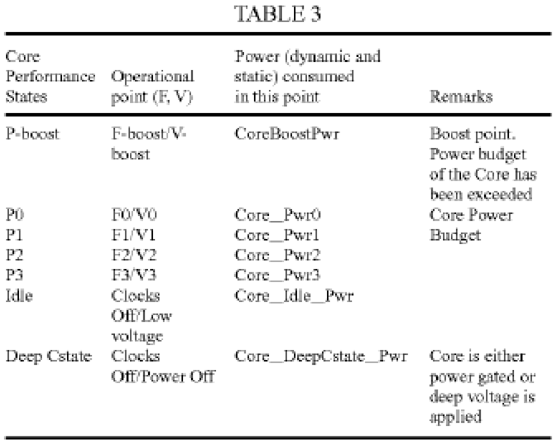
U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	Similarly, *Nussbaum* includes a system on chip (SOC) 100 containing "multiple CPU processing cores 101, a GPU (Graphics Processing Unit) 103, an I/O Bridge 105 (named South-Bridge in some embodiments) and a North-Bridge 107 (which may be combined with the Memory Controller in some embodiments)." *Nussbaum* ¶ [0020]; *see also id.* at Fig. 1.  A processor core operates on a first domain The GPU 103 operates on a second domain *FIG. 1*

<table>
<tr><th><u>U.S. Patent No. 8,769,316</u></th><th><u>Exemplary Disclosures Relevant to <i>Felter</i> and Related Combinations</u></th></tr>
<tr><td></td><td>
<p>In <i>Nussbaum</i>’s SOC 100, processing core 101 operates on a first power domain, and GPU 103 operates on a second power domain. For example, in Table 1, <i>Nussbaum</i> teaches that Core0 of processing cores 101 may operate at a different power than GPU 103:</p>



<p>TABLE 1</p>
<table>
<tr><th>On-die component</th><th>Allocated Power</th></tr>
<tr><td>Core0</td><td>8 w</td></tr>
<tr><td>Core1</td><td>8 w</td></tr>
<tr><td>Core2</td><td>8 w</td></tr>
<tr><td>Core3</td><td>8 w</td></tr>
<tr><td>GPU</td><td>5 w</td></tr>
<tr><td>Memory Controller</td><td>2 w</td></tr>
<tr><td>I/O Bridge</td><td>1 w</td></tr>
<tr><td>Total</td><td>40 w</td></tr>
</table>

<p><i>Nussbaum</i> further teaches that a “processor core may be in one of N performance states . . . characterized by a unique pair of core voltage and frequency values.” <i>Nussbaum</i> ¶ [0033]. Table 3 describes exemplary core power states, including different frequency(F)/voltage (V) value doublets:</p>



<p>TABLE 3</p>
<table>
<tr><th>Core Performance States</th><th>Operational point (F, V)</th><th>Power (dynamic and static) consumed in this point</th><th>Remarks</th></tr>
<tr><td>P-boost</td><td>F-boost/V-boost</td><td>CoreBoostPwr</td><td>Boost point. Power budget of the Core has been exceeded</td></tr>
<tr><td>P0</td><td>F0/V0</td><td>Core_Pwr0</td><td>Core Power Budget</td></tr>
<tr><td>P1</td><td>F1/V1</td><td>Core_Pwr1</td><td></td></tr>
<tr><td>P2</td><td>F2/V2</td><td>Core_Pwr2</td><td></td></tr>
<tr><td>P3</td><td>F3/V3</td><td>Core_Pwr3</td><td></td></tr>
<tr><td>Idle</td><td>Clocks Off/Low voltage</td><td>Core_Idle_Pwr</td><td></td></tr>
<tr><td>Deep Cstate</td><td>Clocks Off/Power Off</td><td>Core_DeepCstate_Pwr</td><td>Core is either power gated or deep voltage is applied</td></tr>
</table>
</td></tr>
</table>

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	See below

The GPU also may be in one of several power states. *See Nussbaum* ¶ [0034]. Table 4 describes exemplary GPU power states:

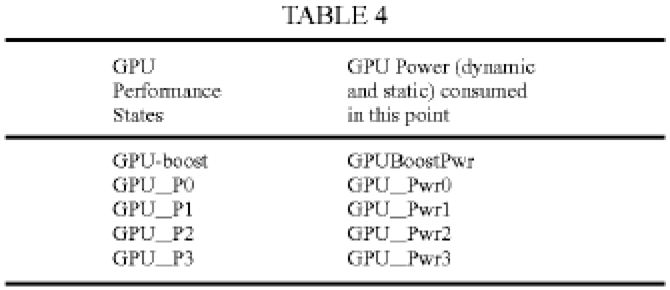
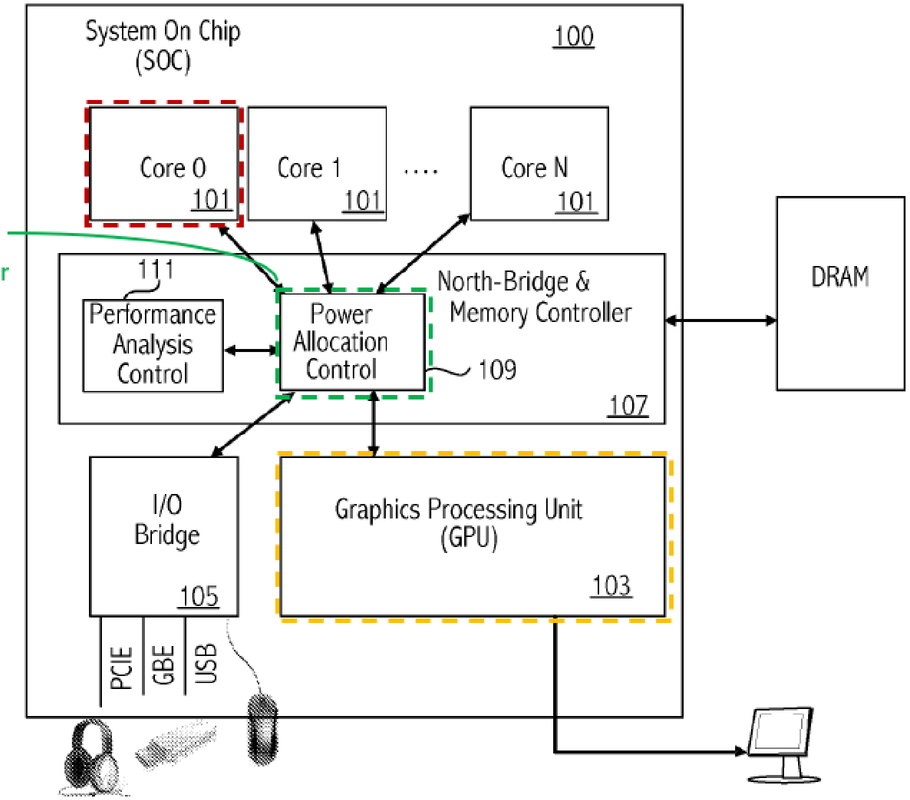


TABLE 4

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

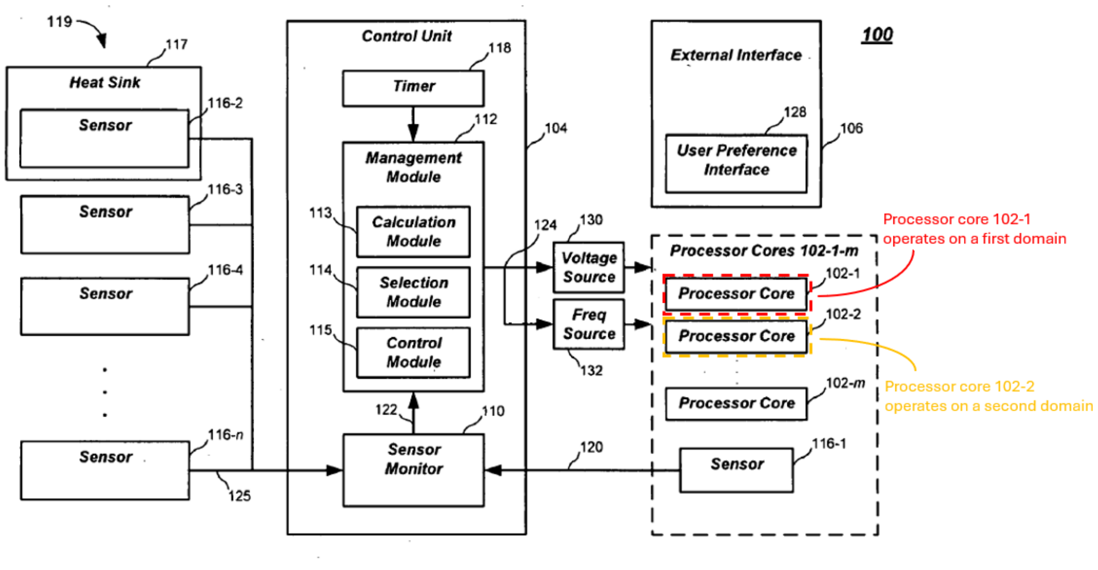
Because processing cores and the GPU operate independently on different power domains, “characterized by a unique pair of core voltage and frequency values,” a POSITA would have understood processing cores and the GPU operate at independent voltages and frequencies. *Nussbaum* ¶ [0033]; *see also id.* ¶ [0037] (explaining “the power of a particular computational unit (voltage×current) is based on the frequency of the clock signal, the supply voltage, and the amount of activity in the computational unit. . .”).

Additionally, *Nussbaum* describes a power allocation controller 109, which “is the functional element that controls allocation of the thermal design point (TDP) power head room to the on-die or on-platform components.” *Id.* ¶ [0020]. Given their similarity in purpose, a POSITA would have understood that *Nussbaum*’s power allocation controller 109 would implement the functionality of *Felter*’s power administrator 200.

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	 Power allocation controller 109 would implement the functionality of power administrator 200 *FIG. 1*

<table>
<tr><th><u>U.S. Patent No. 8,769,316</u></th><th><u>Exemplary Disclosures Relevant to Felter and Related Combinations</u></th></tr>
<tr><td></td><td>

Likewise, *Rotem* includes an apparatus 100, containing multiple processor cores 102-1-*m*. *See Rotem* ¶ [0013], Fig. 1. The elements of apparatus 100 may "be implemented within a single-core or multi-core processor." *Rotem* ¶ [0013], Fig. 5 (depicting processor 502). *Rotem* explains that "management module 112 may control the performance levels for the multiple processor cores 102-1-*m* individually or collectively." *Id.* ¶ [0038]. Given the performance levels of the processor cores may be controlled individually, a POSITA would have understood that each processor core 102-1-*m* may be on a different domain. *See* '316 pat. at 1:54-56 ("[T]he term 'domain' is used to mean a collection of hardware and/or logic that operates at the same voltage and frequency point."). For example, processor core 102-1 may operate on a first domain and processor core 102-2 may operate on a second domain.



Processor core 102-1 operates on a first domain

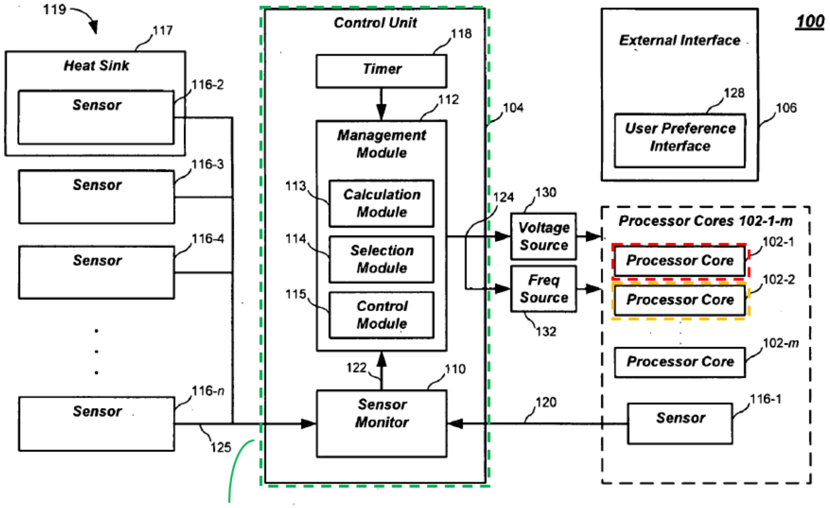
Processor core 102-2 operates on a second domain

FIG. 1

</td></tr>
</table>

<table>
<tr><th>U.S. Patent No. 8,769,316</th><th>Exemplary Disclosures Relevant to *Felter* and Related Combinations</th></tr>
<tr><td></td><td>

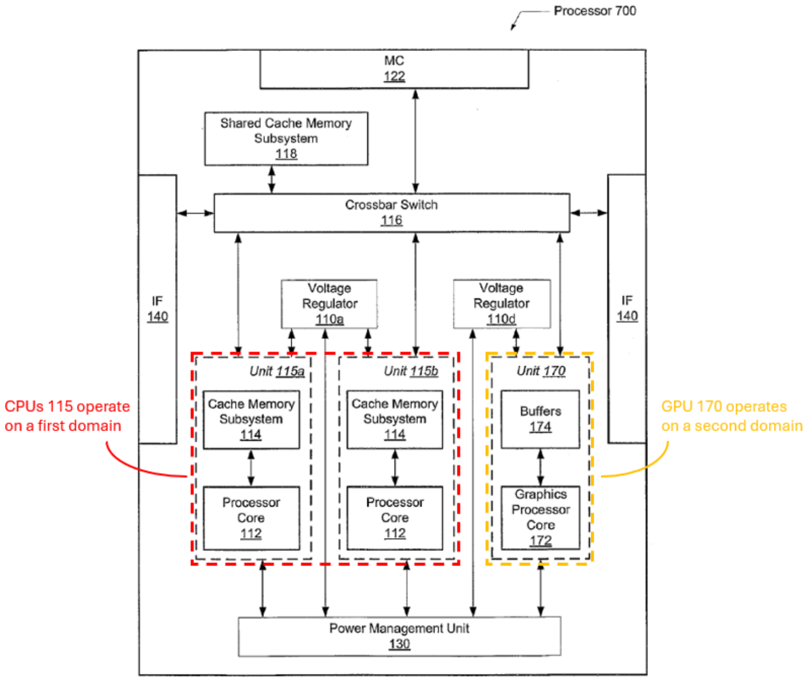
Rotem also describes a control unit 104. *See Rotem* ¶ [0013], ¶ [0018], Fig. 1. Control unit 104 “is arranged to control performance levels for the processor cores 102-1-*m*. This may be accomplished, for example, by the control unit 104 establishing various operational parameters for the processor cores 102-1-*m*.” *Id.* ¶ [0018]. For example, management module 112 “may vary the voltage and/or frequency (e.g., a particular P-state) of the processor cores 102-1-*m* by controlling the corresponding sources 130, 132, which may adjust their output to one or more of the processor cores 102-1-*m* accordingly.” *Id.* ¶ [0026]. Accordingly, a POSITA would have understood that control unit 104 is a power controller because it controls the voltage and/or frequency of various domains, and thus the power of those domains. *See, e.g.*, *Felter* ¶¶ [0037]-[0039] (describing power as a function of voltage and frequency). Given their similarity in purpose, a POSITA would have understood that *Rotem*’s control unit 104 would implement the functionality of *Felter*’s power administrator 200.

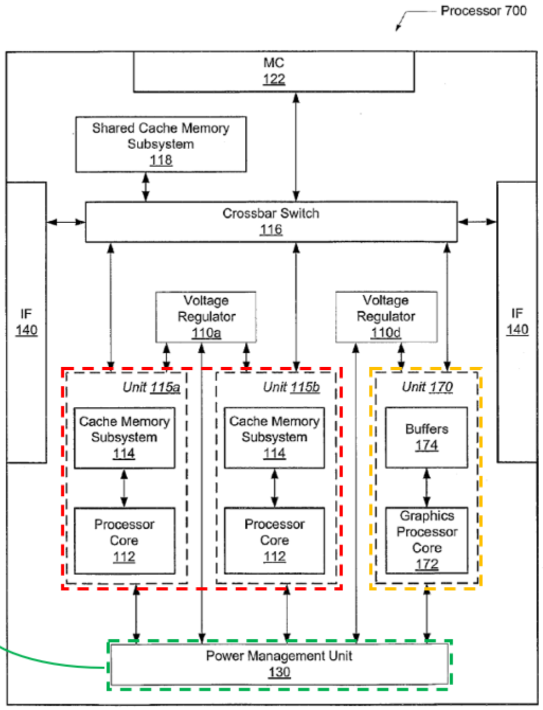


Control unit 104 would implement the functionality of power administrator 200

FIG. 1

</td></tr>
</table>

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	Similarly, *Naffziger* includes a processor 700, containing multiple CPUs 115 and a GPU 170. *See Naffziger* ¶ [0076], Fig. 8. *Naffziger* explains that the CPUs 115 may share the same voltage regulator 110a, but the GPU is coupled to a separate voltage regulator 110d. *See Naffziger* ¶ [0083]. Further, *Naffziger* explains that power may be transferred between the CPUs 115 and the GPU 170, revealing that they are operating at independent power levels. *Id*. Accordingly, since the CPUs 115 and the GPU 170 operate at both independent voltage and independent power levels, a POSITA would have understood that they operate on separate domains. *See, e.g.*, *Felter* ¶¶ [0037]-[0039] (describing power as a function of voltage and frequency).  *FIG. 8*

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	Additionally, *Naffziger* includes a power management unit 130. *See Naffziger* ¶ [0067], Figs. 1, 8. The power management unit 130 allocates power credits to computation units, which determines how much power the computation unit receives. *See Naffziger* ¶ [0024], ¶¶ [0029]-[0031], ¶ [0069]. For example, the power management unit 130 may “transfer power between the CPUs 115 and GPU 170.” *Naffziger* ¶ [0083]. Given their similarity in purpose, a POSITA would have understood that *Naffziger*’s power management unit 130 would implement the functionality of *Felter*’s power administrator 200.  Power management unit 130 would implement the functionality of power administrator 200 *FIG. 8*

U.S. Patent No. 8,769,316	Exemplary Disclosures Relevant to *Felter* and Related Combinations
	Exemplary teachings from the prior art include the following: *Felter* discloses the following at ¶ [0028]: System 100 as depicted in FIG. 1 is only one example of the type of system for which the power management techniques disclosed herein are applicable. The disclosed power management method is also applicable to single-chip systems in which multiple types of components (e.g., processor cores, cache memories, and the like) are integrated within a single piece of silicon, multi-component systems within a single computer (e.g., multi-processor systems with shared or distributed system memory), and multi-machine systems (e.g., multiple uniprocessor or multiprocessor computers within a cluster). *Felter* discloses the following at ¶ [0030]: The performance monitors 201, 206, and 212 and the performance throttles 211, 216, and 222 communicate with a dynamic power allocation administrator 200. Dynamic power allocation administrator 200 (also referred to herein simply as power administrator 200) allocates the available power to system components based on predicted levels of activity for each component. Power administrator 200 also 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. All or part of power administrator 200 may be implemented as computer executable instructions stored on a computer readable medium. In other embodiments, power administrator 200 may be implemented as an integrated or separate on-chip/on-board programmable or hardcoded microcontroller.

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	Felter discloses the following at ¶ [0057]: FIG. 5 depicts additional details of a method 500 for allocating power according to one embodiment. In the depicted embodiment, method 500 includes determining (block 502) the total power budget (P_{BUDGET}) of the system. This operation may require nothing more than retrieving a static value indicating the maximum rating of the system power supply. In other cases, determining the maximum power may include determining the power that the power supply is currently capable of delivering and may take into consideration external factors such as the ambient temperature and the like, possibly in conjunction with a cooperating service processor, management module, or both. *See also Felter* at ¶¶ [0031], [0050]-[0051], [0054], [0058]-[0061], [0064]-[0066]. *Finkelstein* discloses the following at ¶ [0014]: The system 100 may also include a power source 120 (e.g., a direct current (DC) power source or an alternating current (AC) power source) to provide power to one or more components of the system 100. In some embodiments, the power source 120 may include one or more battery packs. The power source 120 may be coupled to components of system 100 through a voltage regulator (VR) 130. Moreover, even though FIG. 1 illustrates one power source 120 and one voltage regulator 130, additional power sources and/or voltage regulators may be utilized. For example, each of the processors 102 may have corresponding voltage regulator(s) and/or power source(s). Also, the 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).

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	Finkelstein discloses the following at ¶ [0016]: As shown in FIG. 1, the processor 102 may further include a power management logic 140 to control supply of power to components of the processor 102 (e.g., cores 106). Logic 140 may have access to one or more storage devices discussed herein to store information relating to operations of logic 140 such as information communicated with various components of system 100 as discussed here. As shown, the logic 140 may be coupled to the VR 130 and/or other components of system 100 (such as the cores 106). For example, the logic 140 may be coupled to receive information (e.g., in the form of one or more bits or signals) to indicate status of one or more sensors 150 (where the sensor(s) 150 may be provided proximate to components of system 100 (or other computing systems discussed herein such as those discussed with reference to other figures including 4 and 5, for example), such as the cores 106, interconnections 104 or 112, etc., to sense variations in temperature, operating frequency, operating voltage, power consumption, inter-core communication activity, etc.) and/or information from one or more power monitoring logics 145 (e.g., which may indicate the operational status of various components of system 100 such as operating temperature, operating frequency, operating voltage, operating status (e.g., active or inactive), power consumption (instantly or over a period of time), etc.). The logic 140 may instruct the VR 130, power source 120, and/or individual components of system 100 (such as the cores 106) to modify their operations. In an embodiment, variations may be sensed in such a way to account for leakage versus active power. For example, logic 140 may indicate to the VR 130 and/or power source 120 to adjust their output. In some embodiments, logic 140 may request the cores 106 to modify their operating frequency, power consumption, etc. Even though components 140, 145, and 150 are shown to be included in processor 102-1, these components may be provided elsewhere in the system 100. For example, power management logic 140 may be provided in the VR

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	130, in the power source 120, directly coupled to the interconnection 104, within one or more (or alternatively all) of the processors 102, etc. *Finkelstein* discloses the following at ¶ [0024]: At an operation 304, it may be determined how many cores in a processor are active and cold (e.g., at a temperature value that is below an excessive threshold temperature value, for example, as detected by the sensor(s) 150). For example, logic 140 may consider statistics (e.g., provided by monitor(s) 145, sensor(s) 150, and/or cores 106 themselves) on operating states of the cores 106 to determine which cores 106 are active and cold. At an operation 306, the information of operations 302 and/or 304 may be taken into account (including various penalties such as those discussed with reference to FIG. 2, including for example, voltage transitions penalty for DVS, etc.), and possible constraints (e.g., we may not desire to decrease the frequency of a hot core too much to maintain smooth operation of critical applications) to determine which technique(s) to use to manage power consumption in a multi-core processor. At an operation 308, the selected technique(s) may be applied to throttle or modify an operational characteristic of one or more of the cores 106 (such as an operating voltage and/or an operating frequency of a processor core 106). For example, logic 140 may consider DVS, pure throttling techniques, or some combinations thereof to control the power consumption of processors 102. Moreover, in one embodiment, operation 308 may apply a combination of the techniques, e.g., where DVS reduces all the cores to a certain power state (P-state), and the hot core is additionally slowed down by one of the pure throttling techniques. *See also Finkelstein* at ¶¶ [0009]-[0011], [0031]-[0032], [0039], [0042], [0049], Figs. 6, 7.

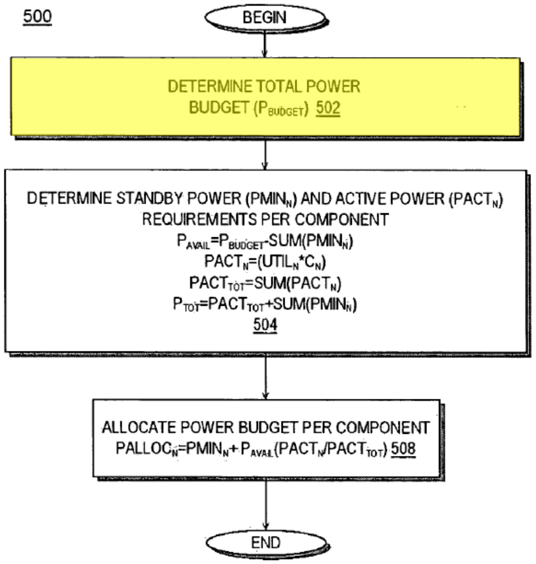
U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	Nussbaum discloses the following at ¶¶ [0020]-[0021]: FIG. 1 shows a high-level view of an exemplary System on a Chip (SOC) 100 incorporating an embodiment of the invention. The SOC 100 includes multiple CPU processing cores 101, a GPU (Graphics Processing Unit) 103, an I/O Bridge 105 (named South-Bridge in some embodiments) and a North-Bridge 107 (which may be combined with the Memory Controller in some embodiments). The power allocation controller 109 is the functional element that controls allocation of the Thermal Design Point (TDP) power headroom to the on-die or on-platform components. The performance analysis control logic 111 analyzes performance sensitivity of the cores and other computational units as described further herein. Note that while the power allocation control 109 and performance analysis center 111 are shown as being part of the North-Bridge 107, in other embodiments they may be located elsewhere in the SOC 100. A TDP (Thermal Design Point) represents the power that can be consumed by the entire SOC and depends on such factors as the form-factor, available cooling solution, AC adapter/battery, and voltage regulator. The SOC performance is optimized within the current TDP and in an embodiment, the power limit corresponding to the TDP is never exceeded. Assume the SOC power limit is the SOC_TDP_Limit. SOC characterization is typically based on allocating maximum power for each of the on-die components while staying within the SOC_TDP_Limit. That occurs by setting the highest operational point (in frequency (F) and voltage (V)) so that even maximally anticipated activity executed at this operational point will not cause the power to exceed the allocated envelope. For example, assume that maximum power of a 4-Core SOC is limited by a 40 w TDP envelope. Table 1 itemizes the power budget allocated for each of the on-die components. *See also Nussbaum* at ¶¶ [0033]-[0034], [0037], [0039], [0047].

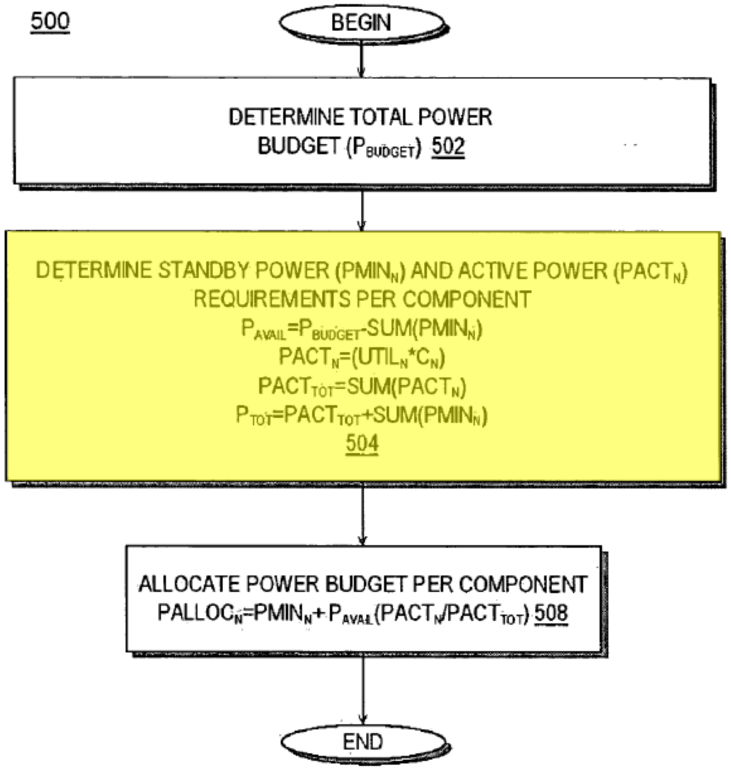
U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	Rotem discloses the following at ¶ [0013]: As shown in FIG. 1, the apparatus 100 may include various elements. For instance, FIG. 1 shows that apparatus 100 may include one or more processor cores 102-1-*m*, a control unit 104, and an external interface 106. Also, apparatus 100 may include various sensors 116-1-*n*. In certain embodiments, the elements of apparatus 100 may be implemented within a single-core or multi-core processor, examples of which are described with reference to FIG. 5. In further embodiments, however, implementations may involve external software and/or external hardware. *Rotem* discloses the following at ¶ [0018]: The control unit 104 is arranged to control performance levels for the processor cores 102-1-m. This may be accomplished, for example, by the control unit 104 establishing various operational parameters for the processor cores 102-1-m. These established operational parameters are based in part on an assessed thermal capacity for the apparatus 100 and/or a computing platform including the apparatus 100. As shown in FIG. 1, control unit 104 includes a sensor monitor 110, a management module 112 and a timer 118. *See also Rotem* at ¶¶ [0021], [0038]-[0039], [0051]. *Naffziger* discloses the following at ¶ [0031]: When a given computation unit does not have a high or moderate workload, its activity level may decrease below a given threshold. Accordingly, its measured power usage value decreases. The resulting reduced power usage value is conveyed to the power management unit 130. In response to the reduced power usage value, the power management unit 130 may redistribute the power credits of the die 102. For example, the power management unit 130 may lend power credits of a determined inactive

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	computation unit to a computation unit that is determined to be highly active. In one embodiment, the power management unit 130 may increase the TDP of a highly active computation unit causing it to maintain or even increase its high activity level. The extra thermal energy generated by this highly active computation unit may be dissipated across the bulk silicon of the die, across the metal on the back of the die, through a heat sink, and through the ambient environment being cooled by a system fan. The relatively inactive computation unit aids in the dissipation of the extra-generated thermal energy allowing the highly active computation unit to maintain high performance. *Naffziger* discloses the following at ¶ [0076]: Referring now to FIG. 8, one embodiment of an exemplary processor 700 is shown. Processor 700 may include memory controller 122, interface logic 140, one or more processing units 115, which may include one or more processor cores 112 and a corresponding cache memory subsystems 114; packet processing logic 116, and a shared cache memory subsystem 118. In addition, processor 700 may include one or more graphics processing units (GPUs) 170. The GPU 170 may comprise a processor core 172 with a parallel architecture, such as a single instruction multiple data (SIMD) core. Examples of SIMD cores include graphics processing units (GPUs), digital signal processing (DSP) cores, and so forth. *Naffziger* discloses the following at ¶ [0083]: As shown in FIG. 8, the CPUs 115 share a same voltage regulator 110a. Alternatively, one or more CPUs 115 may be coupled to a separate voltage regulator. The GPU 170 is coupled to a separate voltage regulator 110d from the CPUs 115. Each of the cores 112 within each CPU 115 and each of the cores 172 within GPU 170 may monitor and measure a corresponding activity level. Each CPU 115 and GPU 170 may aggregate the respective activity levels and report a result to the power management unit 130. This reporting may occur at the end of each given time interval. The power management

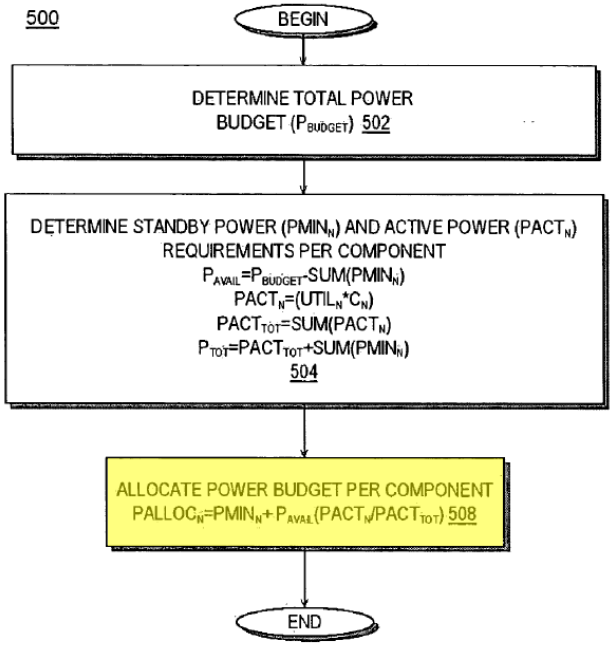
<table>
<tr><th><u>U.S. Patent No. 8,769,316</u></th><th><u>Exemplary Disclosures Relevant to <i>Felter</i> and Related Combinations</u></th></tr>
<tr><td></td><td><blockquote>unit 130 may determine to transfer power between the CPUs 115 and GPU 170 using the methods described earlier.</blockquote>
<i>See also Naffziger</i> at ¶¶ [0021]-[0022], [0024], [0029]-[0030], [0034], [0040]-[0042], [0067]-[0069], [0077]-[0082], Figs. 1, 6.</td></tr>
<tr><td>[8b] determining, in the power controller, a portion of the power budget to be allocated to the first and second domains,</td><td><i>Felter</i>, alone or in combination with one or more references, discloses or suggests "determining, in the power controller, a portion of the power budget to be allocated to the first and second domains."

<i>Felter</i> teaches a method 500 for allocating power. <i>See Felter</i> ¶¶ [0057]-[0061], Fig. 5. Method 500 includes determining a power budget of the system (P<small>BUDGET</small>). <i>See Felter</i> ¶ [0057], Fig. 5 at block 502.


 </td></tr>
</table>

<table>
<tr><th>U.S. Patent No. 8,769,316</th><th>Exemplary Disclosures Relevant to *Felter* and Related Combinations</th></tr>
<tr><td></td><td>


Next, the power controller determines the standby power ($PMIN_N$) and the active power ($PACT_N$) requirements of each component. *See Felter* ¶ [0058], Fig. 5 at block 504. The standby power ($PMIN_N$) represents the minimum drawn by a component. *See Felter* ¶ [0058] (referring to $PMIN_N$ as the “standby or minimum power”), ¶ [0061] (“[T]he 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}.”). The active power ($PACT_N$) represents “the amount of active power per component expected during the upcoming interval,” and is calculated based on the
</td></tr>
</table>

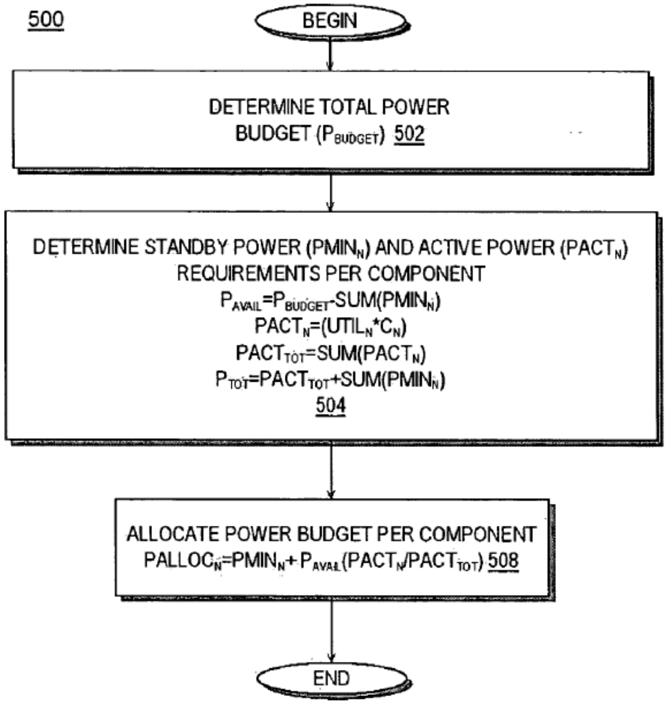
U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	predicted activity of the component. *Felter* ¶ [0059]. For example, $PACT_N$ may be equal to "the product of the activity predicted for component N ($UTIL_N$) and a constant of proportionality for component N (C_N)." *Felter* ¶ [0059], Fig. 5 at block 504. Based on the power budget and standby power requirements, an available power figure (P_{AVAIL}) can be calculated. *See Felter* ¶ [0058], Fig. 5 at block 504. Specifically, $P_{AVAIL} = P_{BUDGET} – SUM(PMIN_N)$. *Id*. Then, the power allocated to each component may be calculated. *See Felter* ¶ [0061], Fig. 5 at block 508. Specifically, the power allocated to component N ($PALLOC_N$) can be determined using the following equation: $PALLOC_N = PMIN_N + P_{AVAIL}(PACT_N/PACT_{TOT})$. *Id*. 

<u>**U.S. Patent No. 8,769,316**</u>	<u>**Exemplary Disclosures Relevant to *Felter* and Related Combinations**</u>
	Accordingly, $PALLOC_1$ would be the power allocated to the first domain and would be calculated as $PALLOC_1 = PMIN_1 + P_{AVAIL}(PACT_1/PACT_{TOT})$. Similarly, $PALLOC_2$ would be the power allocated to the second domain and would be calculated as $PALLOC_2 = PMIN_2 + P_{AVAIL}(PACT_2/PACT_{TOT})$. A POSITA would have found it obvious in each of the alleged combinations for the power controller to implement such functionality. **Exemplary teachings from the prior art include the following:** *Felter* discloses the following at ¶¶ [0057]-[0061]: > FIG. 5 depicts additional details of a method 500 for allocating power according to one embodiment. In the depicted embodiment, method 500 includes determining (block 502) the total power budget (P_{BUDGET}) of the system. This operation may require nothing more than retrieving a static value indicating the maximum rating of the system power supply. In other cases, determining the maximum power may include determining the power that the power supply is currently capable of delivering and may take into consideration external factors such as the ambient temperature and the like, possibly in conjunction with a cooperating service processor, management module, or both. > In block 502, the system determines a set of values required to properly allocate the total power budget. The set of values determined in block 504 includes the standby values (minimum power values) of each system component. Using the standby values, an available power figure (P_{AVAIL}) is determined by subtracting from the maximum power deliverable by the system the sum of all the standby values. Under the assumption that each component always draws its standby or minimum power, P_{AVAIL} represents the amount of power that is available for dynamic allocation.

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	As depicted in FIG. 5, allocating the system power includes using the values of predicted activity generated in block 304 (FIG. 3) to compute the amount of active power per component expected during the upcoming interval. These computations require knowledge of the relationship between the monitored activity and the component's power consumption. The implementation illustrated in FIG. 5 uses a linear model for the processor and memory components. Specifically, 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). Method 500 thus includes determining the expected level of active power consumption required by each component N. From this set of calculations, the total amount active power expected to be required for an upcoming interval ($PACT_{TOT}$) is computed as the sum of the individual component’s expected active power. Having thus determined the total expected active power and the total amount of standby power required, method 500 includes determining the total required power (P_{TOT}) for all components as the sum of the total active power $PACT_{TOT}$ and the total standby power required $SUM(PMIN_N)$. Having determined the power budget and the active and standby demands of each of the components, the depicted embodiment of method 500 includes allocating (block 508) power to the system components. In the embodiment depicted in FIG. 5, for example, power is allocated to the various components used a modified pro rata technique. More specifically, 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}. 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 $PALLOC_N = PMIN_N + P_{AVAIL}(PACT_N/PACT_{TOT})$. This formula 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)

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	executing on the system. The formula expressed in block 508 represents a technique for distributing scarce system power equitably so that the system performance of each component is impacted proportionately. *See also Felter* at ¶¶ [0051], [0054], [0062]-[0065], Fig. 3.
[8c] including allocating a minimum reservation value to the first domain and a minimum reservation value to the second domain,	*Felter*, alone or in combination with one or more references, discloses or suggests “including allocating a minimum reservation value to the first domain and a minimum reservation value to the second domain.” As explained above, *Felter* teaches that the power allocated to a domain N ($PALLOC_N$) can be calculated with the following equation: $PALLOC_N = PMIN_N + P_{AVAIL}(PACT_N/PACT_{TOT})$. *See supra* [8b]; *Felter* ¶ [0061], Fig. 5. $PMIN_N$ is the standby, or minimum power of domain N. *See Felter* ¶ [0058] (“The set of values determined in block 504 includes the standby values (minimum power values) of each system component.”). The standby power for the domain is always allocated to each domain. *See Felter* ¶ [0058] (“Under the assumption that each component always draws its standby or minimum power . . . .”), ¶ [0061] (“[T]he 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}.”), ¶ [0010] (“Allocating the power budget 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.”). Accordingly, the standby power is the minimum reservation value of a domain. Specifically, $PMIN_1$ is the minimum reservation value of the first domain and $PMIN_2$ is the minimum reservation value of the second domain.

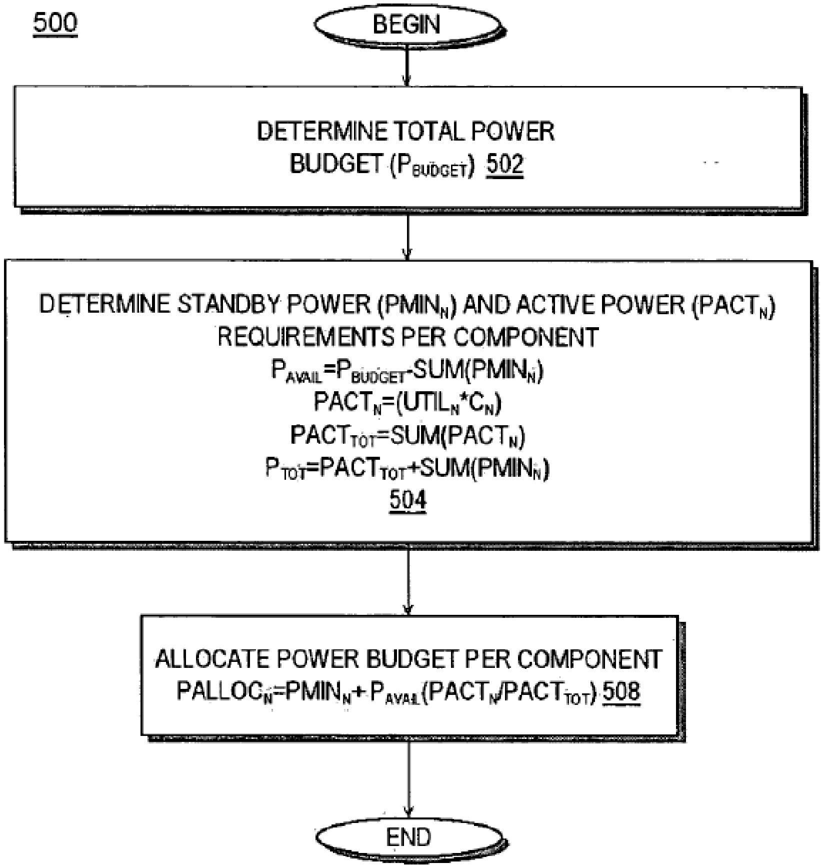
U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	Exemplary teachings from the prior art include the following: *Felter* discloses the following at ¶ [0058]: In block 502, the system determines a set of values required to properly allocate the total power budget. The set of values determined in block 504 includes the standby values (minimum power values) of each system component. Using the standby values, an available power figure (P_{AVAIL}) is determined by subtracting from the maximum power deliverable by the system the sum of all the standby values. Under the assumption that each component always draws its standby or minimum power, P_{AVAIL} represents the amount of power that is available for dynamic allocation. *Felter* discloses the following at ¶ [0061]: Having determined the power budget and the active and standby demands of each of the components, the depicted embodiment of method 500 includes allocating (block 508) power to the system components. In the embodiment depicted in FIG. 5, for example, power is allocated to the various components used a modified pro rata technique. More specifically, 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}. 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 $PALLOC_N = PMIN_N + P_{AVAIL}(PACT_N/PACT_{TOT})$. This formula 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. The formula expressed in block 508 represents a technique

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	for distributing scarce system power equitably so that the system performance of each component is impacted proportionately. *Felter* discloses the following at Fig. 5:  *See also Felter* at ¶¶ [0010], [0065].

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
[8d] and 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*, alone or in combination with one or more references, discloses or suggests “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.” As explained above, *Felter* teaches determining a total power budget (P_{BUDGET}). *See supra* [8a]; *Felter* ¶ [0057], Fig. 5 at block 502. Additionally, *Felter* explains that each domain has a minimum reservation value (e.g., its standby power $PMIN_N$). *See supra* [8c]; *Felter* ¶ [0058], Fig. 5 (block 504). *Felter* also teaches that an available power figure (P_{AVAIL}) can be calculated by subtracting the sum of all the standby values from the power budget. *See Felter* ¶ [0058], Fig. 5 (block 504). Accordingly, P_{AVAIL} is the remaining portion of the power budget. “P_{AVAIL} represents the amount of power that is available for dynamic allocation.” *Felter* ¶ [0058]. P_{AVAIL} is distributed based on expected levels of activity. *See Felter* ¶ [0010]. Specifically, 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.” *Felter* ¶ [0061]. This is represented in the power allocation equation: $PALLOC_N = PMIN_N + P_{AVAIL}(PACT_N/PACT_{TOT})$. *Id*. Accordingly, the portion of the remaining portion of the power budget (P_{AVAIL}) allocated to a domain N is determined according to a sharing policy value, $PACT_N$, representing the expected level of active power consumption of domain N. *See Felter* ¶¶ [0060]-[0061]. The sharing policy value for the first domain would be $PACT_1$ and the sharing policy value for the second domain would be $PACT_2$. **Exemplary teachings from the prior art include the following:** *Felter* discloses the following at ¶¶ [0057]-[0061]: FIG. 5 depicts additional details of a method 500 for allocating power according to one embodiment. In the depicted embodiment, method 500 includes determining (block 502) the total power budget (P_{BUDGET}) of the system. This operation may require nothing more than retrieving a static value indicating the maximum rating of the system power supply. In other cases, determining the maximum power may include

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	determining the power that the power supply is currently capable of delivering and may take into consideration external factors such as the ambient temperature and the like, possibly in conjunction with a cooperating service processor, management module, or both. In block 502, the system determines a set of values required to properly allocate the total power budget. The set of values determined in block 504 includes the standby values (minimum power values) of each system component. Using the standby values, an available power figure (P_{AVAIL}) is determined by subtracting from the maximum power deliverable by the system the sum of all the standby values. Under the assumption that each component always draws its standby or minimum power, P_{AVAIL} represents the amount of power that is available for dynamic allocation. As depicted in FIG. 5, allocating the system power includes using the values of predicted activity generated in block 304 (FIG. 3) to compute the amount of active power per component expected during the upcoming interval. These computations require knowledge of the relationship between the monitored activity and the component's power consumption. The implementation illustrated in FIG. 5 uses a linear model for the processor and memory components. Specifically, 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). Method 500 thus includes determining the expected level of active power consumption required by each component N. From this set of calculations, the total amount active power expected to be required for an upcoming interval ($PACT_{TOT}$) is computed as the sum of the individual component’s expected active power. Having thus determined the total expected active power and the total amount of standby power required, method 500 includes determining the total required power (P_{TOT}) for all components as the sum of the total active power $PACT_{TOT}$ and the total standby power required SUM($PMIN_N$).

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	Having determined the power budget and the active and standby demands of each of the components, the depicted embodiment of method 500 includes allocating (block 508) power to the system components. In the embodiment depicted in FIG. 5, for example, power is allocated to the various components used a modified pro rata technique. More specifically, 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}. 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 $PALLOC_N = PMIN_N + P_{AVAIL}(PACT_N/PACT_{TOT})$. This formula 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. The formula expressed in block 508 represents a technique for distributing scarce system power equitably so that the system performance of each component is impacted proportionately.

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	Felter discloses the following at Fig. 5:  *See also Felter* at ¶¶ [0010], [0030], [0066].

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
[8e] controlling a frequency of the first domain and a frequency of the second domain based on the allocated portions.	*Felter*, alone or in combination with one or more references, discloses or suggests “controlling a frequency of the first domain and a frequency of the second domain based on the allocated portions.” *Felter* explains that based on the allocated portions of the power budget, an activity limit can be set for each component. *See Felter* ¶ [0010], ¶ [0062]. If the component’s activity exceeds the activity limit, the component would be throttled. *See Felter* ¶ [0010] (“[I]f the component’s activity attempts to exceed the component’s corresponding activity limit, it is constrained from doing so.”), ¶ [0019] (“If the level of activity of a component exceeds its limit, the system will throttle or otherwise constrain the activity for the corresponding component.”) ¶ [0020] (“The system includes hardware or software to monitor the activity proxies and to throttle them when an activity level exceeds its limit.”). *Felter* teaches that one method for throttling component activity is to limit the number of instructions dispatched for a processor over a specified time interval and to halt the instruction dispatch if the limit is reached (i.e., “pipeline throttling”). *See Felter* ¶ [0033]. However, *Felter* also explains that, as an alternative to pipeline throttling, voltage and frequency scaling can be used in devices that support it. *See Felter* ¶ [0035]. By “reducing the voltage in conjunction with the frequency, one can obtain a cubic (with respect to frequency) reduction in active power.” *Felter* ¶ [0036]. A POSITA would have understood that the devices taught by *Finkelstein*, *Nussbaum*, *Rotem*, and *Naffziger* would have all supported voltage and frequency scaling because they all disclose adjusting the voltage and frequency to manage power. *See e.g., Finkelstein* ¶ [0028] (“In some embodiments, energy budget may be managed and/or power setting(s) (e.g., voltage and/or frequency changes) may be made accordingly to a current budget.”); *Nussbaum* ¶ [0032] (“[A] frequency sensitivity table is maintained as a result of the frequency sensitivity training for all components whose performance can be adjusted, typically through adjusting frequency (and voltage if necessary).”); *Rotem* ¶ [0001] (“[H]eat generated by a processor may be limited by controlling an operating frequency and operating voltage for the processor . . .”), ¶ [0026] (“The operational performance of the processor cores 102-1-*m* may be controlled in part by the voltage source 130 and/or the frequency source 132.”); *Naffziger* ¶ [0064] (“P-state throttling may in turn be associated with a decrease in the operational frequency f and the operational Voltage V.”). Accordingly, *Felter* teaches controlling the voltage and frequency to limit power usage

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	(according to the activity limit set based on the allocated portions of the power budget), which is “a more efficient technique” than pipeline throttling. *Felter* ¶ [0036]. **Exemplary teachings from the prior art include the following:** *Felter* discloses the following at ¶ [0010]: The objective identified above is achieved with a method for managing power in a data processing system having multiple components as disclosed herein. The method includes determining a power budget for the system. Activity levels during a forthcoming time interval are then predicted for each of the components. Using the predicted activity levels, the power budget is allocated among the system components. An activity limit is then established for each component based on its corresponding portion of the power budget. The activity of a component is then monitored and, if the component's activity attempts to exceed the component's corresponding activity limit, it is constrained from doing so. Determining the predicted level of activity may include determining a predicted number of instructions dispatched by a processor component or a predicted number of memory requests serviced for a system memory component. Allocating the power budget 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. Monitoring an activity limit may include monitoring a processor performance monitor configured to count the number of instructions dispatched during a timing interval. In this embodiment, constraining the activity of the processor may include pipeline throttling in which the processor is prevented from dispatching additional instructions until the current timing interval expires. Alternatively, constraining processor activity may be achieved by reducing the voltage and frequency applied to the processor.

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	Felter discloses the following at ¶¶ [0035]-[0036]: For devices supporting sophisticated power reduction techniques such as voltage and frequency scaling, the method for increasing performance by dynamically allocating the power budget of a system among multiple components is equally applicable when using activity regulation mechanisms other than pipeline throttling. What is needed is a clear understanding of the power and performance tradeoffs for different levels of the regulation mechanism and the workload requirements. This can be obtained by good modeling, empirical characterization or a combination of both. One example of an alternative activity regulation technique is voltage-frequency scaling. Compared to pipeline throttling, voltage-frequency scaling may be a more efficient technique because reducing the voltage in conjunction with the frequency, one can obtain a cubic (with respect to frequency) reduction in active power. In addition, reducing the voltage for the circuits also reduces key components of the static/leakage power, further improving efficiency. However, dynamic voltage and frequency scaling (DVFS) is often a more complex regulation mechanism to implement because of twin controls of voltage and frequency and, so, changes initiated by it can be slower to take effect than with pipeline throttling *See also Felter* at ¶¶ [0019]-[0020], [0033]-[0034], [0037]-[0049], [0062]. *Finkelstein* discloses the following at ¶ [0028]: In some embodiments, energy budget may be managed and/or power setting(s) (e.g., voltage and/or frequency changes) may be made accordingly to a current budget. Energy-based power management may be performed by controlling the energy budget defined iteratively as follows:

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	$E_{n+1}=\alpha E_n+(TDP_n-P_n)\Delta t_n$ *See also Finkelstein* at ¶¶ [0003], [0016], [0019], [0024], [0042], [0048]. *Nussbaum* discloses the following at ¶ [0032]: The Boost Sensitivity Table (BST) is maintained as a result of a frequency sensitivity training session for the components to be potentially boosted. In other embodiments, a frequency sensitivity table is maintained as a result of the frequency sensitivity training for all components whose performance can be adjusted, typically through adjusting frequency (and voltage if necessary). In an embodiment, power budget reallocation uses the information in the BST to decide which on-die component(s) are the most sensitive to boosting and thus “deserve” to get a higher TDP power margin reallocated when a reallocation takes place. *See also Nussbaum* at ¶¶ [0033]-[0035], [0037], [0047]. *Rotem* discloses the following at ¶ [0001]: Modern computing systems generate heat during operation. The heat may affect certain platform components of a system, and is therefore generally required to be limited, and such limitations may affect operations of one or more processors. Heat generated by the computing system may be limited using various dynamic thermal management (DTM) techniques. For example, heat generated by a processor may be limited by controlling an operating frequency and operating voltage for the processor, or utilizing a heat sink attached to the processor. Further, various platform-level cooling devices may be implemented for the computing system to perform heat dissipation, such as heat pipes, heat links, heat transfers, heat spreaders, vents, fans, blowers, and liquid based coolants.

U.S. Patent No. 8,769,316	Exemplary Disclosures Relevant to *Felter* and Related Combinations
	Rotem discloses the following at ¶ [0026]: In various embodiments, for example, the management module 112 may send the signal 124 as a voltage level signal or a frequency level signal to a respective voltage source 130 or a frequency source 132. The operational performance of the processor cores 102-1-*m* may be controlled in part by the voltage source 130 and/or the frequency source 132. For example, the voltage source 130 may comprise a variable regulator and the frequency source 132 may comprise a voltage controlled oscillator (VCO), among other components. The management module 112 may vary the voltage and/or frequency (e.g., a particular P-state) of the processor cores 102-1-*m* by controlling the corresponding sources 130, 132, which may adjust their output to one or more of the processor cores 102-1-*m* accordingly. Alternatively or additionally, the management module 112 may adjust clock toggling settings for the processor cores 102-1-*m*. *See also Rotem* at ¶¶ [0015], [0021], [0023], [0032], [0034]-[0035], [0041]-[0045], [0048]-[0049]. *Naffziger* discloses the following at ¶ [0064]: As seen in FIG. 5, although transferring power credits may yield an overall performance boost for a chip, monitoring particular conditions may be performed to avoid exceeding design constraints. For example, after a given component receives power credits, at a later time, the workload may change due to changes caused by software applications. An increased workload may cause two terms (α, C) in the power consumption expression for the given component to increase. As is well known, the power consumption of integrated circuits is proportional to αfCV^2, wherein α represents a switching factor, f represents frequency, C represents capacitance, and V represents voltage. Generally speaking, a higher workload causes the switching factor α and the capacitance C to increase. Accordingly, the given component may begin P-State throttling in response to the increased power consumption. Generally, should power

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	consumption increase beyond a given threshold, the P-state may be throttled in order to maintain a relatively stable power consumption value. P-state throttling may in turn be associated with a decrease in the operational frequency f and the operational voltage V. *See also Naffziger* at ¶¶ [0006], [0009], [0023], [0032]-[0036], [0048]-[0050], [0062]-[0063].
Claim 9	
[9] The method of claim 8, wherein	*Felter*, alone or in combination with one or more references, discloses or suggests the method of claim 8. *See supra* claim 8.
determining the portion of the power budget to be allocated to the first and second domains includes allocating the power budget to the second domain and not to the first domain if the power budget is less than the minimum reservation value for the second domain.	Claims 9 through 11 are invalid under 35 U.S.C. § 112, first paragraph, because they are incompatible with claim 8, from which they depend. Additionally, under Daedalus’s apparent view of the claims, they are invalid as anticipated and/or obvious in view of one or more prior art references. Under Daedalus’s apparent view of the claims, *Felter*, alone or in combination with one or more references, discloses or suggests “determining the portion of the power budget to be allocated to the first and second domains includes allocating the power budget to the second domain and not to the first domain if the power budget is less than the minimum reservation value for the second domain.” If the power budget is less than the minimum reservation value of the second domain, then it would be impossible for the system to allocate enough power to each domain to satisfy both minimum reservation values. A POSITA would have understood the system must determine how to allocate the available power budget under such circumstances.

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	In this scenario, there would be only two possibilities: (1) allocate all of the power budget to one domain or (2) split the power budget between the domains. In *KSR Int'l Co. v. Teleflex Inc.*, the Supreme Court held that when "there is a design need . . . to solve a problem and there are a finite number of identified, predictable solutions, a person of ordinary skill has good reason to pursue the known options within his or her technical grasp." 550 U.S. 398, 421 (2007). Here, a POSITA would have understood there is a design need to allocate a power budget when there is not enough power to meet the minimum reservation requirements, and there are only two "identified, predictable solutions." *Id*. Two solutions is a "finite number." For example, in *Uber Techs., Inc. v. X One, Inc.*, the Federal Circuit found there were only two possible methods for displaying a map with plotted locations: "server-side plotting and terminal-side plotting." 957 F.3d 1334, 1339 (Fed. Cir. 2020). Because of this, the Federal Circuit determined there were only a "finite number of identified, predictable solutions," and determined the claims at issue were obvious. *Id*.; *see also Google LLC v. Koninklijke Philips N.V.*, 795 F. App'x 840, 841 (Fed. Cir. 2020) (non-precedential); *Cardiovalve Ltd. v. Edwards Lifesciences Corp.*, No. 2022-2230, 2024 WL 1208638, at *2 (Fed. Cir. Mar. 21, 2024) (non-precedential). Here, as in *Uber*, there are only two possible solutions: assign all the power to one domain or split it between the domains. Likewise, in *Perfect Web Techs., Inc. v. InfoUSA, Inc.*, Perfect Web asserted a patent for "managing bulk e-mail distribution to groups of targeted consumers." 587 F.3d 1324, 1326 (Fed. Cir. 2009). Step (D) of representative claim 1 required repeating steps (A)-(C) if the calculated quantity of emails does not exceed a prescribed minimum quantity of successfully received e-mails. *Id.* The Federal Circuit only identified three possible solutions for the problem step (D) attempts to solve: "(1) oversending, or e-mailing an excess of addresses to ensure the quota is met; (2) if some addresses failed or "bounced" back messages, re-sending to those same addresses in the hope that a second transmission somehow succeeds; and (3) identifying a new group of addresses and sending messages to them, which is step (D)." *Id*. at 1331. The Federal Circuit concluded there were only a "finite number of identified, predictable solutions," i.e., three solutions. *Id*. Accordingly, the method proposed by Perfect Web in

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	step (D) “would have been obvious.” *Id*. because two solutions is fewer than three solutions, it follows from *Perfect Web* that two possible solutions would also represent a “finite number of identified, predictable solutions.” *Id*. Because two solutions is fewer than three solutions, it follows from *Perfect Web* that two possible solutions would also represent a “finite number of identified, predictable solutions.” *Id*. A POSITA would have needed to account for all possible power supply conditions, including the claimed conditions, when designing a system. Although a system may lack sufficient power to obtain a minimum amount of work from either domain under the claimed conditions, a POSITA would have understood a domain may draw power for other purposes over the next interval, not just performing work. This includes, for instance, maintaining domain cache memories and volatile device contexts. *See, e.g.*, Advanced Configuration and Power Interface (ACPI) Specification, Rev. 4.0, at 32, 35, 38-41, 46, 51, 298-311, 333, 483-89 (June 16, 2009). Otherwise, the domain may need to be put to sleep. *See id*. Here, when considering the claimed circumstances, a POSITA would have understood one of the domains (e.g., the second domain) should receive a *de minimis* amount of power to avoid putting it to sleep, thereby avoiding the attendant delays caused by moving volatile cache memory states and device contexts into non-volatile memory (and back again once the domain wakes up). *See id.* Because putting one domain to sleep rather than two improves system responsiveness, a POSITA would have found it obvious under the claimed conditions to allocate all the power budget to one of the domains (e.g., the second domain) to avoid putting it to sleep.
Claim 10	
[10] The method of claim 9, further comprising	*Felter*, alone or in combination with one or more references, discloses or suggests the method of claim 9. *See supra* [9].

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
obtaining the minimum reservation value for the second domain from a configuration register written by user-level software.	Claims 9 through 11 are invalid under 35 U.S.C. § 112, first paragraph, because they are incompatible with claim 8, from which they depend. Additionally, under Daedalus’s apparent view of the claims, they are invalid as anticipated and/or obvious in view of one or more prior art references. Under Daedalus’s apparent view of the claims, *Felter*, alone or in combination with one or more references, discloses or suggests “obtaining the minimum reservation value for the second domain from a configuration register written by user-level software.” As explained above, a POSITA would have found it obvious to obtain the minimum reservation value for the second domain (PMIN2) from a configuration register written by user-level software. *See supra* **Exemplary Reasons for Obtaining a Minimum Reservation Value from a Configuration Register Written by User-Level Software in *Finkelstein*’s Power Allocation Method**. **Exemplary teachings from the prior art include the following:** *Felter* discloses the following at ¶ [0058]: In block 502, the system determines a set of values required to properly allocate the total power budget. The set of values determined in block 504 includes the standby values (minimum power values) of each system component. Using the standby values, an available power figure (P_{AVAIL}) is determined by subtracting from the maximum power deliverable by the system the sum of all the standby values. Under the assumption that each component always draws its standby or minimum power, P_{AVAIL} represents the amount of power that is available for dynamic allocation. *Felter* discloses the following at ¶ [0065]:

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	Alternatively, a reservoir for minimal component activity can be included in the ‘standby’ power allocated for the component to ensure there is always enough power for each component to obtain some minimal work from it. *See also Felter* at ¶¶ [0010], [0037]-[0039], [0057], [0059]-[0061]. *Brinks* discloses the following at ¶ [0002]: Embodiments of the invention generally relate to controlling power consumption of electrical systems. More particularly, an aspect of an embodiment of the invention relates to an intelligent power controller for a complex electrical system, including System on a Chip, through power and clock distributions. *Brinks* discloses the following at ¶¶ [0048]-[0051]: The power management policy 404 maps the different system operating states to different operating points 406 according to a desired user experience, for instance, to maximize energy savings, or as another example, to maximize system reactivity. A power management policy 404 will decide when and how the system shall transition from one operating point to another. The power manager may include an execution engine to implement a chosen policy. The power management policies may be stored in a table, hierarchical table, register, matrix or some combination of the three that map different system operating states to different operating points 406 to improve and/or maximize energy savings for a given policy. The policies may be configured and/or programmed into the table, register, or matrix by a chip designer. The chip designer may run multiple simulations to perfect knowing which components affect other components when the chip runs and how much power is saved. The chip designer may make a matrix of power saving to be referenced

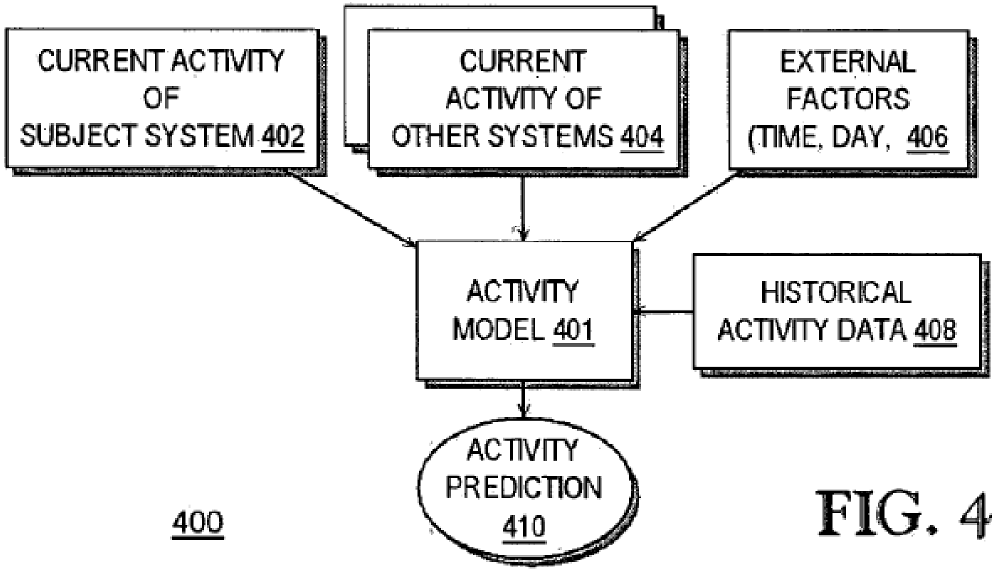
U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	> and which components can be powered off for small granular time periods depending on which application is running. > The power management policies may be initially set after the simulations, but may also be updated during device operation based on actual user feedback or additions of new applications executing on the integrated circuit. Accordingly, the table, register, or matrix may be programmable, so that its behavior can be changed at run-time to support different chip operating modes. Therefore, multiple power management policies may be programmed and or updated by a chip designer at design time, run time, or after fabrication and by a user during operation. . . . > The policy defines the tradeoffs between user experience and power consumption. . . . An example power management policy 404 may favor faster performance over battery life, or vice-versa. *Brinks* discloses the following at ¶ [0061]: > Also as generally described above, the power management policies may be stored in a table, hierarchical table, register, matrix or some combination of the three that map different system operating states to different operating points to maximize energy savings for a given policy. The table, register, matrix may further be programmable, so that its behavior can be changed at run-time to support different chip operating modes so that the multiple power management policies can be programmed and or updated by a chip designer at design time, run time, or after fabrication and by a user during operation. Permitting the chip designer to implement different tradeoffs between performance and power consumption may be used to differentiate a device manufacturer over its competitors. *See also Brinks* at ¶¶ [0022], [0046]-[0047], [0052]-[0060], [0062]-[0066], [0071].

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	See also Finkelstein at ¶¶ [0031], [0034]. *See also Nussbaum* at ¶¶ [0020]-[0021], [0025], [0032]-[0033], [0036]. *See also Rotem* at ¶¶ [0008]-[0009], [0027]-[0028], [0032]-[0034], [0049]. *See also Naffziger* at ¶¶ [0006], [0022], [0024], [0030], [0037].
Claim 11	
[11] The method of claim 9, further comprising	*Felter*, alone or in combination with one or more references, discloses or suggests the method of claim 9. *See supra* [9].
allocating the minimum reservation value for the second domain to the second domain, and allocating a remaining portion of the power budget to the first domain when the power budget is greater than the minimum reservation value for the second domain but less than a sum of the minimum reservation value for the second domain and	Claims 9 through 11 are invalid under 35 U.S.C. § 112, first paragraph, because they are incompatible with claim 8, from which they depend. Additionally, under Daedalus's apparent view of the claims, they are invalid as anticipated and/or obvious in view of one or more prior art references. Under Daedalus's apparent view of the claims, *Felter*, alone or in combination with one or more references, discloses or suggests "allocating the minimum reservation value for the second domain to the second domain, and allocating a remaining portion of the power budget to the first domain when the power budget is greater than the minimum reservation value for the second domain but less than a sum of the minimum reservation value for the second domain and the minimum reservation value for the first domain." If the power budget is less than the sum of the minimum reservation value of the second domain and the minimum reservation value of the first domain, then it would be impossible for the system to

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
the minimum reservation value for the first domain.	allocate enough power to each domain to meet both minimum reservation values. A POSITA would have understood the system must determine how to allocate the available power budget under such circumstances. In this scenario, there would be three ways to allocate the power budget: (1) all of the power budget could be allocated to one domain, (2) the minimum reservation value of one domain may be allocated to that domain and the remaining available power budget may be allocated to the other domain, or (3) the power budget may be split between the two domains such that neither domain is allocated enough power to meet its minimum reservation value. In *KSR Int’l Co. v. Teleflex Inc.*, the Supreme Court held that when “there is a design need . . . to solve a problem and there are a finite number of identified, predictable solutions, a person of ordinary skill has good reason to pursue the known options within his or her technical grasp.” 550 U.S. 398, 421 (2007). Here, there are only three possible solutions, which is a finite number. For example, in *Perfect Web Techs., Inc. v. InfoUSA, Inc.*, Perfect Web asserted a patent for “managing bulk e-mail distribution to groups of targeted consumers.” 587 F.3d 1324, 1326 (Fed. Cir. 2009). Step (D) of representative claim 1 required repeating steps (A)-(C) if the calculated quantity of emails does not exceed a prescribed minimum quantity of successfully received e-mails. *Id.* The Federal Circuit only identified three possible solutions for the problem step (D) attempts to solve: “(1) oversending, or e-mailing an excess of addresses to ensure the quota is met; (2) if some addresses failed or ‘bounced’ back messages, re-sending to those same addresses in the hope that a second transmission somehow succeeds; and (3) identifying a new group of addresses and sending messages to them, which is step (D).” *Id.* at 1331. The Federal Circuit concluded there were only a “finite number of identified, predictable solutions,” i.e., three solutions. *Id.* Accordingly, the method proposed by Perfect Web in step (D) “would have been obvious.” *Id.* A POSITA would have needed to account for all possible power supply conditions when designing the system, including the claimed circumstances. Although the system may lack sufficient power to obtain the minimum amount of work from the first domain under the claimed circumstances, a POSITA would have understood that the first domain may draw power for other purposes over the

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	next interval, too, not just performing work. This includes, for instance, maintaining domain cache memories and volatile device contexts. *See, e.g.*, Advanced Configuration and Power Interface (ACPI) Specification, Rev. 4.0, at 32, 35, 38-41, 46, 51, 298-311, 333, 483-89 (June 16, 2009). Otherwise, the first domain may need to be put into a sleep state. *See id*. A POSITA thus would have found it obvious under the claimed conditions (i.e., when power budget P_{BUDGET} is such that $PMIN_2 < P_{BUDGET} < PMIN_2 + PMIN_1$) to allocate the minimum reservation value for the second domain (e.g., $PMIN_2$) to the second domain, thereby allowing the second domain to perform a minimum level of work, and to allocate the remaining portion of the power budget (i.e., $P_{BUDGET} – PMIN_2$) to the first domain to avoid putting the first domain into a sleep state, which would slow the system. *See supra* [9]; *Felter* ¶ [0065]. **Exemplary teachings from the prior art include the following:** *Felter* discloses the following at ¶ [0065]: Alternatively, a reservoir for minimal component activity can be included in the ‘standby’ power allocated for the component to ensure there is always enough power for each component to obtain some minimal work from it.
Claim 12	
[12] The method of claim 8, wherein	*Felter*, alone or in combination with one or more references, discloses or suggests the method of claim 8. *See supra* [8pre]-[8e].

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
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*, alone or in combination with one or more references, discloses or suggests “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.” As explained above, $PACT_N$ represents the sharing policy value of domain N. *See supra* [8d]; *Felter* ¶¶ [0059]-[0060], Fig. 5. $PACT_N$ represents the “amount of active power . . . . expected during the upcoming interval” for domain N. *Felter* ¶ [0059]. $PACT_N$ is calculated “as the product of the activity (UTIL) predicted for component N ($UTIL_N$) and a constant of proportionality for component N (C_N).” *Id*. A POSITA would have understood that if a request for a higher frequency was not granted, the amount of expected power, based on the predicted activity, was too low. *See, e.g., Nussbaum* ¶ [0037] (“[T]he power of a particular computational unit (voltage × current) is based on the frequency of the clock signal, the supply voltage, and the amount of activity in the computational unit.”). The expected activity in subsequent intervals, and thus the corresponding $PACT_N$ value, would increase. For example, *Felter* explains that the system may predict activity using a behavior model 400. *See Felter* ¶ [0051], Fig. 4.

U.S. Patent No. 8,769,316	Exemplary Disclosures Relevant to *Felter* and Related Combinations
	 FIG. 4 The behavior model 400 receives inputs from current activity input 402, representing “the current activity levels of the subject component.” *Felter* ¶ [0052]. The behavior model 400 may use the current activity input 402 to “generate an activity prediction 410 that is used to develop a power allocation for the corresponding component.” *Felter* ¶ [0053]. Accordingly, a POSITA would have understood that if the current activity was higher than had been predicted for an interval, and a request for a higher frequency was not granted, the behavior model 400 would compute a higher activity prediction, and thus a higher $PACT_N$ value for the next interval. *See Felter* ¶¶ [0052]-[0053], ¶ [0059]. **Exemplary teachings from the prior art include the following:**

U.S. Patent No. 8,769,316	Exemplary Disclosures Relevant to *Felter* and Related Combinations
	Felter discloses the following at ¶¶ [0052]-[0053]: Referring to FIG. 4 momentarily, a block diagram of selected elements of a behavior model 400 includes an activity modeling module 401 that receives input in the form of current activity inputs 402 and 404, external factors 406, and historical activity data 408. Current activity inputs 402 indicates the current activity levels of the subject component (the component for which activity is being predicted) whereas activity inputs 404 represents the activity levels of other components in the dynamic power allocation domain. External factors 406 may include factors such as the time of day, day of week, and the like. Historical activity data 408, as its name suggests, includes archived activity data for one or more of the system components. Activity model 401 processes all or some of these inputs to generate an activity prediction 410 that is used to develop a power allocation for the corresponding component. Activity model may incorporate substantially any prediction technique or prediction algorithm that is compatible with the timing requirements of power administrator 200 (i.e., the prediction algorithm should not create a bottle neck for power administrator 200). *Felter* discloses the following at ¶¶ [0059]-[0060]: As depicted in FIG. 5, allocating the system power includes using the values of predicted activity generated in block 304 (FIG. 3) to compute the amount of active power per component expected during the upcoming interval. These computations require knowledge of the relationship between the monitored activity and the component’s power consumption. The implementation illustrated in FIG. 5 uses a linear model for the processor and memory components. Specifically, 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).

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	Method 500 thus includes determining the expected level of active power consumption required by each component N. From this set of calculations, the total amount active power expected to be required for an upcoming interval ($PACT_{TOT}$) is computed as the sum of the individual component's expected active power. Having thus determined the total expected active power and the total amount of standby power required, method 500 includes determining the total required power (P_{TOT}) for all components as the sum of the total active power $PACT_{TOT}$ and the total standby power required $SUM(PMIN_N)$. *See also Felter* at ¶¶ [0050]-[0051], [0057]-[0058], [0061], Figs. 3-5. **Additional Combination with *Bose-602*:** As explained above, a POSITA would have found it obvious to increment the sharing policy values when a request for a higher frequency is not granted. *See supra* **Exemplary Reasons for Incrementing the Sharing Policy Value when a Request for a Higher Frequency is not Granted**. **Exemplary teachings from the prior art include the following:** *Felter* discloses the following at ¶ [0060]-[0061]: Method 500 thus includes determining the expected level of active power consumption required by each component N. From this set of calculations, the total amount active power expected to be required for an upcoming interval ($PACT_{TOT}$) is computed as the sum of the individual component's expected active power. Having thus determined the total expected active power and the total amount of standby power required, method 500 includes determining the total required power (P_{TOT}) for all components as the sum of the total active power $PACT_{TOT}$ and the total standby power required $SUM(PMIN_N)$.

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	Having determined the power budget and the active and standby demands of each of the components, the depicted embodiment of method 500 includes allocating (block 508) power to the system components. In the embodiment depicted in FIG. 5, for example, power is allocated to the various components used a modified pro rata technique. More specifically, 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}. 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 $PALLOC_N = PMIN_N + P_{AVAIL}(PACT_N/PACT_{TOT})$. This formula 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. The formula expressed in block 508 represents a technique for distributing scarce system power equitably so that the system performance of each component is impacted proportionately. *See also Felter* at ¶¶ [0037]-[0039], [0057]-[0059]. *Bose-602* discloses the following at ¶ [0013]: According to an exemplary embodiment of the present invention, a method of power management of a system of connected components includes initializing a token allocation map across the connected components, wherein each component is assigned a power budget as determined by a number of allocated tokens in the token allocation map, monitoring utilization sensor inputs and command state vector inputs, determining, at first periodic time intervals, a current performance level, a current power consumption level and an assigned power budget for the system based on the utilization sensor inputs and the command state vector inputs, and determining, at second periodic time intervals, a token re-allocation map based on the current

U.S. Patent No. 8,769,316	**Exemplary Disclosures Relevant to *Felter* and Related Combinations**
	performance level, the current power consumption level and the assigned power budget for the system, according to a re-assigned power budget of at least one of the connected components, while enforcing a power consumption limit based on a total number of allocated tokens in the system. *Bose-602* discloses the following at ¶ [0086]: Alternatively, if the resource is below power and temperature limits, its performance level is gauged in block 1314. If it is assessed to be below a predefined minimum performance, block 1316 is used to try and read additional tokens from the centralized bus token storage; such extra tokens are used to boost up performance in the targeted unit in block 1317 (e.g., by increasing its voltage and frequency, if allowed by the system architecture) and then block 1315 is again entered to check if the resource is within the power and temperature limits.