



- (51) **International Patent Classification:**
G06F 1/00 (2006.01) *G06F 1/26* (2006.01)
- (21) **International Application Number:**
PCT/US2012/055943
- (22) **International Filing Date:**
18 September 2012 (18.09.2012)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
13/247,580 28 September 2011 (28.09.2011) US
- (71) **Applicant (for all designated States except US):** **INTEL CORPORATION** [US/US]; 2200 Mission College Boulevard, Santa Clara, California 95052 (US).
- (72) **Inventors:** **ANANTHAKRISHNAN, Avinash N.**; 2900 NW Adagio Way, Hillsboro, Oregon 97124 (US). **ZIV, Tomer**; #5 Pnina & Moshe Street, 75357 Rishon Lezion, M (IL). **RAJWAN, Doron**; 13 Odem Street, 00000 Rishon Le-Zion, M (IL). **ROTEM, Efraim**; 8 Vizo St., 34400 Haifa, HA (IL).
- (74) **Agent:** **ROZMAN, Mark J.**; Trop, Pruner & Hu, P.C., 1616 S. Voss Rd., Ste. 750, Houston, Texas 77057-2631 (US).
- (81) **Designated States (unless otherwise indicated, for every kind of national protection available):** AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM,

DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

- (84) **Designated States (unless otherwise indicated, for every kind of regional protection available):** ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

- as to the identity of the inventor (Rule 4.17(i))
- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))
- as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))

Published:

- without international search report and to be republished upon receipt of that report (Rule 48.2(g))

(54) **Title:** CONTROLLING TEMPERATURE OF MULTIPLE DOMAINS OF A MULTI-DOMAIN PROCESSOR

(57) **Abstract:** In one embodiment, the present invention includes a method for determining, in a controller of a multi-domain processor, whether a temperature of a second domain of the multi-domain processor is greater than a sum of a throttle threshold and a cross-domain margin, and if so, reducing a frequency of a first domain of the multi-domain processor by a selected amount. In this way, a temperature of the second domain can be allowed to reduce, given a thermal coupling of the domains. Other embodiments are described and claimed.



Controlling Temperature Of Multiple Domains Of A Multi-Domain Processor

Background

[0001] As technology advances in the semiconductor field, devices such as processors incorporate ever-increasing amounts of circuitry. Over time, processor designs have evolved from a collection of independent integrated circuits (ICs), to a single integrated circuit, to multicore processors that include multiple processor cores within a single IC package. As time goes on, ever greater numbers of cores and related circuitry are being incorporated into processors and other semiconductors.

[0002] Multicore processors are being extended to include additional functionality by incorporation of other functional units within the processor. Typically, a multicore processor has a common power budget and a common thermal budget. The power budget is set so that a specified power level, at least as averaged over time, is not exceeded. The thermal budget is set such that a thermal throttle point, which is a highest allowable temperature at which the processor can safely operate, is not exceeded. Although these common budgets exist, mechanisms to adaptively share the budgets across the wide variety of circuitry present in a processor does not exist.

Brief Description of the Drawings

[0003] FIG. 1 is a flow diagram of a method in accordance with an embodiment of the present invention.

[0004] FIG. 2 is a block diagram of a processor in accordance with an embodiment of the present invention.

[0005] FIG. 3 is a block diagram of a multi-domain processor in accordance with another embodiment of the present invention.

[0006] FIG. 4 is a block diagram of a system in accordance with an embodiment of the present invention.

Detailed Description

[0007] In various embodiments, a processor having multiple independent domains can be controlled to prevent a temperature of any of the domains from exceeding a maximum junction temperature of the processor. As used herein, the term “maximum junction temperature” is a highest temperature at which a semiconductor product is specified to be fully operational. As an example, this temperature can be determined during device characterization, both during manufacture as well as testing in a laboratory environment, and stored in a non-volatile storage of the device. In general, various circuitry of a processor or other product may be present to provide a throttling mechanism such that the processor is prevented from operating above this maximum junction temperature.

[0008] Examples described herein are in connection with a multicore processor including multiple processor cores and one or more other processing engines. For example, in a particular embodiment described herein, at least two independent domains may be present, namely a core domain that includes one or more cores and a graphics domain that includes one or more graphics engines. As used herein the term “domain” is used to mean a collection of hardware and/or logic that operates at the same voltage and frequency point. As an example, a multicore processor can further include other non-core processing engines such as fixed function units, graphics engines, and so forth. Such processor can include at least two independent domains, one associated with the cores (referred to herein as a core domain) and one associated with a graphics engine (referred to herein as a graphics domain). Although many implementations of a multi-domain processor can be formed on a single semiconductor die, other implementations can be realized by a multi-chip package in which different domains can be present on different semiconductor die of a single package.

[0009] In various embodiments, cross-domain effects with regard to temperature, such that an increased temperature of one domain has an effect on the temperature of another domain can be considered and taken into account in controlling temperature of each domain. More specifically, embodiments may provide a programmable temperature margin parameter, referred to herein as a “cross-domain

margin,” that is used to determine a marginal temperature level above a throttle temperature on a first domain at which a power consumption level of a different domain is to be reduced in order to allow the first domain temperature to begin cooling. This cross-domain margin may apply even if the power consumption of the different domains does not change over time. As with the throttle points, the cross-domain margin can be stored in a non-volatile storage of the processor.

[0010] Embodiments may be performed in various locations. As one example, logic of a power control unit (PCU) of a processor can be used to perform the thermal analysis and temperature control in accordance with an embodiment of the present invention. Referring now to FIG. 1, shown is a flow diagram of a method in accordance with an embodiment of the present invention. Method 100 of FIG. 1 may be performed by thermal logic of a PCU, in one embodiment, and can be performed on an iterative basis, namely once per predetermined interval (which may be approximately 1 millisecond (ms) in one embodiment). In general, the method may be performed based on a temperature of multiple domains of a processor. Such temperature information can be obtained directly from the domains, e.g., from a thermal sensor associated with each domain that is coupled to the PCU, e.g., via a push bus. Or, temperature information can be determined within the PCU, e.g., based on thermal sensor data obtained from these thermal sensors.

[0011] Method 100 may begin by determining the appropriate domain under review. Thus at diamond 110 it can be determined whether the domain under review is a core domain. The embodiment of FIG. 1 is for a multi-domain processor including two domains, namely a core domain that includes one or more processor cores and a non-core domain, which in a particular embodiment can be a graphics domain that includes one or more graphics engines. Of course, this method can be extended to a product having more than two domains that can share a common thermal (and power) budget. If it is determined that the domain under review is a core domain, control passes to diamond 120.

[0012] At diamond 120 it can be determined whether the temperature of the core domain is greater than a throttle point. Although the scope of the present invention is not limited in this regard, in some embodiments this throttle point may correspond

to a maximum junction temperature (or T_j) at which a given domain of the processor is specified to be fully operational without breakdown. Note that this T_j can be fixed, e.g., in non-volatile storage or fuse logic. However, this throttle point can be configurable, e.g., by software or firmware to a value lower than T_j . In this way, an original equipment manufacturer (OEM) can dial down the throttle point as a function of T_j (e.g., using basic input/output system (BIOS)). As an example and not for purposes of limitation, for a multicore processor the throttle point for a core domain can be set at between approximately 80 and 110 degrees Celsius (and may be set between approximately the same or a different range for a non-core domain). In some embodiments, this throttle point can be determined by testing, e.g., during fabrication, and stored in a non-volatile storage or other mechanism of the processor. However, in other embodiments the throttle point can be dynamically changed, e.g., based on a history of the semiconductor product such that the throttle point can be reduced as the product ages due to various degradation mechanisms of the semiconductor product.

[0013] Note that this throttle point can be reached at different performance levels of a processor. For example, according to an operating system (OS)-based mechanism, namely the Advanced Configuration and Platform Interface (ACPI) standard (e.g., Rev. 3.0b, published October 10, 2006), a processor can operate at various performance states or levels, namely from P0 to PN. In general, the P1 performance state may correspond to the highest guaranteed performance state that can be requested by an OS. In addition to this P1 state, the OS can further request a higher performance state, namely a P0 state. This P0 state may thus be an opportunistic state in which, when power and/or thermal budget is available, processor hardware can configure the processor or at least portions thereof to operate at a higher than guaranteed frequency. In many implementations a processor can include multiple so-called bin frequencies above a guaranteed maximum frequency, also referred to as a P1 frequency. At any of these performance states, and more likely in a P0 or P1 state, the throttle point may be reached.

[0014] If it is determined that the core domain temperature is greater than this throttle point, control passes to block 140 where a frequency of the core domain can be reduced by a selected amount. This reduction in frequency in turn causes the power consumption of the domain also to decrease, leading to a decrease in the domain's temperature. As one example, the selected amount may correspond to a degradation of a predetermined amount of a frequency bin. For example, the core domain frequency can be reduced by $1/N$ of a bin frequency. As used herein, a "bin frequency" corresponds to a smallest multiple by which a domain frequency can be updated. In some embodiments this bin frequency can be an integer multiple of a bus clock frequency, although the scope of the present invention is not limited in this regard.

[0015] Note that in many implementations, rather than performing frequency reductions by less than a bin frequency amount, embodiments allow for multiple iterations of the frequency control algorithm of FIG. 1 to be performed, and a count maintained of the number of $1/N$ reductions. Then at a natural bin boundary, which corresponds to N/N reductions, an actual frequency reduction to the given domain can be performed. In general, this frequency reduction can be performed using frequency control logic of the PCU, which can cause the given domain to have its frequency reduced. As an example, this frequency reduction process includes stopping all useful work of the domain, draining transactions pending in the domain, applying the new frequency (and possibly a new voltage) to the domain, and then resuming normal operation in that domain at the new frequency/voltage. Thus as seen, certain complexity is involved in a frequency change and furthermore this frequency change can take some amount of time. Temperature change can lag a frequency change and the amount of such time lag is a function of the thermal capacitance of the cooling solution deployed. By controlling the rate of frequency change to match the thermal capacitance of the cooling solution, control loop oscillations and fluctuations in frequency and temperature can be avoided. Accordingly, by only performing such frequency changes upon reaching a natural bin boundary, greater efficiency can be realized. This count of iterations at which a frequency is to be reduced by a $1/N$ amount can be stored, e.g., in a temporary storage such as a register, counter or other such storage. When the value in this

counter thus reaches an integral amount, an actual frequency change can be performed.

[0016] Note that the mechanism to reduce the core domain frequency can be performed in different manners. For example, frequency control logic of the PCU can receive an instruction to update the core domain frequency. In turn, the frequency control logic may select various instructions to be sent to cause the frequency to be reduced. For example, various control signals can be sent to one or more phase lock loops (PLLs) or other frequency control mechanisms to cause the frequency to be reduced. After this frequency change, control passes to block 180, where a check for temperature of another domain can begin.

[0017] If instead at diamond 120 it is determined that the core domain temperature is not greater than the throttle point, control passes next to diamond 130, where it can be determined whether the graphics domain temperature is greater than a sum of the throttle point (for the graphics domain) and a cross-domain margin. Thus as discussed above, a temperature of one domain can affect the temperature of another domain. And accordingly, if the graphics domain temperature is greater than this sum of throttle point and cross-domain margin, the same adjustment to the core domain frequency can be performed, namely a reduction by a selected amount occurs at block 140. Of course, a different amount of reduction can be effected in this case, e.g., $1/X$ of a bin frequency. Note that both X and N are independent parameters that can be stored in non-volatile storage.

[0018] Otherwise, there is no potential thermal violation and accordingly the current frequency of the core domain can be maintained. Thus at this point, control passes to diamond 150, where an analysis with regard to the graphics domain temperature can be made. As seen, this determination can also be reached if the domain under review is the graphics domain. Thus at diamond 150 it can be determined whether the graphics domain temperature is greater than a throttle point. If so, control passes to block 170 where the graphics domain frequency can be reduced by a selected amount. Note that this reduction in graphics domain frequency can proceed as discussed above with regard to block 140. For example, the graphics frequency can be reduced by $1/N$ of a bin frequency. Note that the different domains can have

different bin frequencies. Control thereafter passes back to block 180 discussed above.

[0019] Still referring to FIG. 1, if instead it is determined that the graphics domain temperature is not greater than the throttle point, control passes from diamond 150 to diamond 160 where it can be determined whether the core domain temperature is greater than a sum of the throttle point and the cross-domain margin. If so, the graphics domain frequency can be reduced at block 170. Otherwise, control passes back for further analysis of the core domain at diamond 120.

[0020] Thus as seen, FIG. 1 proceeds in an iterative manner where both domain temperatures are iteratively analyzed. Although shown with this particular implementation in the embodiment of FIG. 1, understand the scope of the present invention is not limited in this regard.

[0021] Note that the value of N can be a tunable or programmable parameter that determines the rate of cooling achieved by thermal throttling in accordance with an embodiment of the present invention. For a small value of N (e.g., less than approximately 2), the rate at which the frequency can be reduced is faster than for a larger value of N. However, such smaller N values may lead to oscillations in which a domain can be overcooled. In some embodiments, the value of N may be programmable, e.g., by a user using a user-level instruction. In other embodiments, the value of N can be configured via a configuration register or stored in a non-volatile storage, e.g., of the PCU.

[0022] Thus in a multi-domain processor, embodiments can enable thermal control such that not only does a domain that is operating above its thermal specification have its temperature reduced, but at least one other domain within the same die can have its temperature reduced. In this way, the contribution to thermal heating of the domain under consideration by another domain can be reduced or removed.

[0023] Referring now to FIG. 2, shown is a block diagram of a processor in accordance with an embodiment of the present invention. As shown in FIG. 2, processor 300 may be a multicore processor including a plurality of cores 310_a – 310_n. In one embodiment, each such core may be of an independent power domain

and can be configured to operate at an independent voltage and/or frequency, and to enter a turbo frequency mode when available headroom exists. The various cores may be coupled via an interconnect 315 to a system agent or uncore 320 that includes various components. As seen, the uncore 320 may include a shared cache 330 which may be a last level cache. In addition, the uncore may include an integrated memory controller 340, various interfaces 350 and a power control unit 355.

[0024] In various embodiments, power control unit 355 may include a thermal control logic 359, which may be a logic to control domain frequencies based on temperature not only on the domain to be controlled, but also other domains of the processor. In the embodiment of FIG. 2, assuming that each core is of an independent power domain, logic 359 can dynamically determine whether a temperature has exceeded a throttle point and/or a sum of a throttle point and cross-domain margin, and if so to perform frequency control in accordance with an embodiment of the present invention. As further seen in FIG. 2 to provide for storage of different values for use in thermal control, a thermal control storage 357 may further be present within PCU 355 to store values such as the T_j and/or throttle points, cross-domain margin, and N . Although shown at this location in the embodiment of FIG. 2, understand that the scope of the present invention is not limited in this regard and the storage of this information can be in other locations, such as configuration registers, non-volatile storage or the like.

[0025] With further reference to FIG. 2, processor 300 may communicate with a system memory 360, e.g., via a memory bus. In addition, by interfaces 350, connection can be made to various off-chip components such as peripheral devices, mass storage and so forth. While shown with this particular implementation in the embodiment of FIG. 2, the scope of the present invention is not limited in this regard.

[0026] Referring now to FIG. 3, shown is a block diagram of a multi-domain processor in accordance with another embodiment of the present invention of the present invention. As shown in the embodiment of FIG. 3, processor 400 includes multiple domains. Specifically, a core domain 410 can include a plurality of cores 410a–410n, a graphics domain 420 can include one or more graphics engines, and a

system agent domain 450 may further be present. In various embodiments, system agent domain 450 may execute at a fixed frequency and may remain powered on at all times to handle power control events and power management such that domains 410 and 420 can be controlled to dynamically change frequency based on thermal events occurring on one or the other domain in accordance with an embodiment of the present invention. Each of domains 410 and 420 may operate at different voltage and/or power. Note that while only shown with three domains, understand the scope of the present invention is not limited in this regard and additional domains can be present in other embodiments. For example, multiple core domains may be present each including at least one core.

[0027] In general, each core 410 may further include low level caches in addition to various execution units and additional processing elements. In turn, the various cores may be coupled to each other and to a shared cache memory formed of a plurality of units of a last level cache (LLC) 440₀ – 440_n. In various embodiments, LLC 450 may be shared amongst the cores and the graphics engine, as well as various media processing circuitry. As seen, a ring interconnect 430 thus couples the cores together, and provides interconnection between the cores, graphics domain 420 and system agent circuitry 450.

[0028] In the embodiment of FIG. 3, system agent domain 450 may include display controller 452 which may provide control of and an interface to an associated display. As further seen, system agent domain 450 may include a power control unit 455 which can include a frequency control logic 459 in accordance with an embodiment of the present invention. In various embodiments, this logic may execute algorithms such as shown in FIG. 1 to thus dynamically control frequency based on thermal events occurring on at least one of the core and graphics domains.

[0029] As further seen in FIG. 3, processor 400 can further include an integrated memory controller (IMC) 470 that can provide for an interface to a system memory, such as a dynamic random access memory (DRAM). Multiple interfaces 480₀ – 480_n may be present to enable interconnection between the processor and other circuitry. For example, in one embodiment at least one direct media interface (DMI) interface may be provided as well as one or more Peripheral Component Interconnect

Express (PCI Express™ (PCIe™)) interfaces. Still further, to provide for communications between other agents such as additional processors or other circuitry, one or more interfaces in accordance with a Intel® Quick Path Interconnect (QPI) protocol may also be provided. Although shown at this high level in the embodiment of FIG. 3, understand the scope of the present invention is not limited in this regard.

[0030] Embodiments may be implemented in many different system types. Referring now to FIG. 4, shown is a block diagram of a system in accordance with an embodiment of the present invention. As shown in FIG. 4, multiprocessor system 500 is a point-to-point interconnect system, and includes a first processor 570 and a second processor 580 coupled via a point-to-point interconnect 550. As shown in FIG. 4, each of processors 570 and 580 may be multicore processors, including first and second processor cores (i.e., processor cores 574a and 574b and processor cores 584a and 584b), although potentially many more cores may be present in the processors. Each of the processors can include a PCU or other logic to perform dynamic frequency control of a domain of the processor based on thermal events of multiple domains, as described herein.

[0031] Still referring to FIG. 4, first processor 570 further includes a memory controller hub (MCH) 572 and point-to-point (P-P) interfaces 576 and 578. Similarly, second processor 580 includes a MCH 582 and P-P interfaces 586 and 588. As shown in FIG. 4, MCH's 572 and 582 couple the processors to respective memories, namely a memory 532 and a memory 534, which may be portions of system memory (e.g., DRAM) locally attached to the respective processors. First processor 570 and second processor 580 may be coupled to a chipset 590 via P-P interconnects 552 and 554, respectively. As shown in FIG. 4, chipset 590 includes P-P interfaces 594 and 598.

[0032] Furthermore, chipset 590 includes an interface 592 to couple chipset 590 with a high performance graphics engine 538, by a P-P interconnect 539. In turn, chipset 590 may be coupled to a first bus 516 via an interface 596. As shown in FIG. 4, various input/output (I/O) devices 514 may be coupled to first bus 516, along with a bus bridge 518 which couples first bus 516 to a second bus 520. Various devices

may be coupled to second bus 520 including, for example, a keyboard/mouse 522, communication devices 526 and a data storage unit 528 such as a disk drive or other mass storage device which may include code 530, in one embodiment. Further, an audio I/O 524 may be coupled to second bus 520. Embodiments can be incorporated into other types of systems including mobile devices such as a smart cellular telephone, tablet computer, netbook, or so forth.

[0033] Embodiments may be implemented in code and may be stored on a non-transitory storage medium having stored thereon instructions which can be used to program a system to perform the instructions. The storage medium may include, but is not limited to, any type of disk including floppy disks, optical disks, solid state drives (SSDs), compact disk read-only memories (CD-ROMs), compact disk rewritables (CD-RWs), and magneto-optical disks, semiconductor devices such as read-only memories (ROMs), random access memories (RAMs) such as dynamic random access memories (DRAMs), static random access memories (SRAMs), erasable programmable read-only memories (EPROMs), flash memories, electrically erasable programmable read-only memories (EEPROMs), magnetic or optical cards, or any other type of media suitable for storing electronic instructions.

[0034] While the present invention has been described with respect to a limited number of embodiments, those skilled in the art will appreciate numerous modifications and variations therefrom. It is intended that the appended claims cover all such modifications and variations as fall within the true spirit and scope of this present invention.

What is claimed is:

- 1 1. A processor comprising:
2 a first domain including at least one core to execute instructions;
3 a second domain including at least one functional unit to execute a
4 specialized function, wherein the first and second domains can operate at
5 independent frequencies; and
6 a power control unit (PCU) coupled to the first and second domains, the PCU
7 including a thermal logic to cause a reduction in a frequency of the first domain if a
8 temperature of the second domain exceeds a sum of a throttle threshold and a
9 cross-domain margin.
- 1 2. The processor of claim 1, wherein the cross-domain margin corresponds to a
2 marginal level above the throttle threshold at which a power consumption of the first
3 domain is to be reduced to allow a temperature of the second domain to be reduced.
- 1 3. The processor of claim 1, wherein the thermal logic is to cause the reduction
2 in the first domain frequency by an amount corresponding to $1/N$ of a frequency bin
3 of the processor.
- 1 4. The processor of claim 3, wherein the thermal logic is to update a value of a
2 counter for the first domain when the second domain is determined to exceed the
3 sum.
- 1 5. The processor of claim 4, wherein the thermal logic is to cause the first
2 domain frequency to be reduced by the frequency bin when the counter value is
3 equal to N .
- 1 6. The processor of claim 3, wherein the thermal logic is to update a value of a
2 counter for the first domain when the first domain is determined to exceed a throttle
3 threshold for the first domain.

1 7. The processor of claim 3, wherein the first domain frequency reduction of $1/N$
2 comprises an update to a counter.

1 8. The processor of claim 3, where N is controllable by a user.

1 9. The processor of claim 1, wherein the thermal logic is to cause the reduction
2 in the first domain frequency if the first domain temperature is greater than a throttle
3 threshold for the first domain.

1 10. The processor of claim 1, wherein the thermal logic is to cause a reduction in
2 a frequency of the second domain if a temperature of the first domain exceeds a sum
3 of a throttle threshold for the first domain and the cross-domain margin.

1 11. The processor of claim 1, wherein the processor includes a non-volatile
2 storage to store the throttle threshold and the cross-domain margin, wherein the
3 throttle threshold is of a different value than a junction temperature of the processor.

1 12. A method comprising:
2 determining, in a controller of a multi-domain processor, whether a
3 temperature of a second domain of the multi-domain processor is greater than a sum
4 of a throttle threshold and a cross-domain margin; and
5 if so, reducing a frequency of a first domain of the multi-domain processor by
6 a selected amount, otherwise maintaining a current frequency of the first domain.

1 13. The method of claim 12, further comprising reducing the first domain
2 frequency if a temperature of the first domain is greater than the throttle threshold.

1 14. The method of claim 12, further comprising updating a value of a counter for
2 the first domain when the second domain is determined to exceed the sum, and
3 updating the counter value when the first domain is determined to exceed a throttle
4 threshold for the first domain.

1 15. The method of claim 14, wherein reducing the first domain frequency includes
2 reducing the first domain frequency by a bin frequency of the first domain when the
3 counter value equals N.

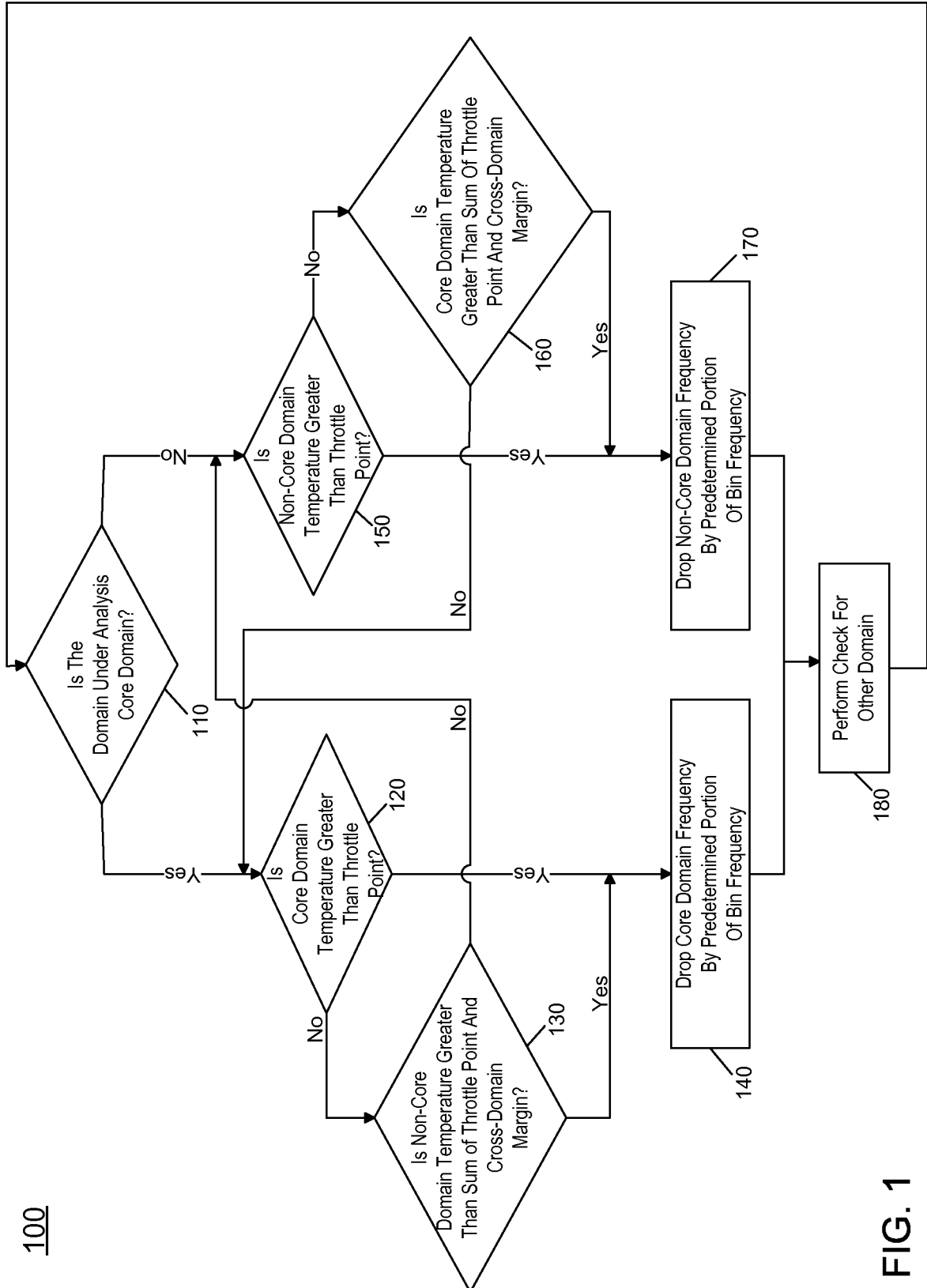
1 16. The method of claim 12, further comprising:
2 determining whether a temperature of the first domain is greater than the sum;
3 and
4 if so, reducing a frequency of the second domain by a second selected
5 amount, otherwise maintaining a current frequency of the second domain.

1 17. A system comprising:
2 a multi-domain processor including a core domain having a plurality of cores,
3 a non-core domain including at least one processing engine, and a system agent
4 domain including a power controller, wherein the power controller is to reduce a
5 frequency of the core domain by a selected amount when a temperature of the non-
6 core domain is greater than a sum of a throttle threshold and a cross-domain margin,
7 and otherwise to maintain a current frequency of the core domain.

1 18. The system of claim 17, wherein the power controller is to reduce the
2 frequency of the core domain after a plurality of iterations in which it is determined
3 that the non-core domain temperature is greater than the sum.

1 19. The system of claim 17, wherein the power controller is to reduce the
2 frequency of the core domain after a plurality of iterations in which it is determined
3 that the non-core domain temperature is greater than the throttle threshold.

1 20. The system of claim 17, wherein the power controller is to update a value of a
2 counter for the core domain when the non-core domain is determined to exceed the
3 sum, and to reduce the core domain frequency by a bin frequency when the counter
4 value equals N.



300

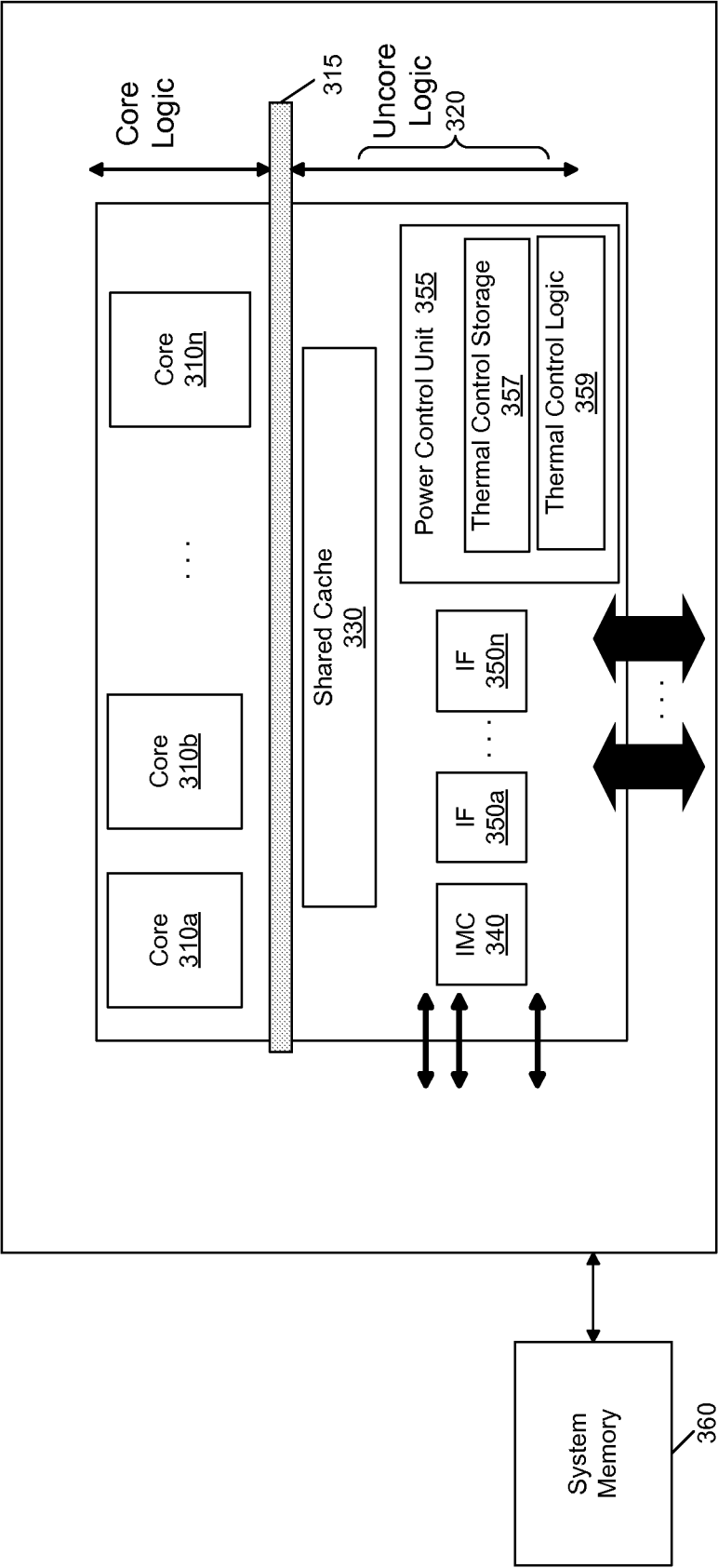


FIG. 2

400

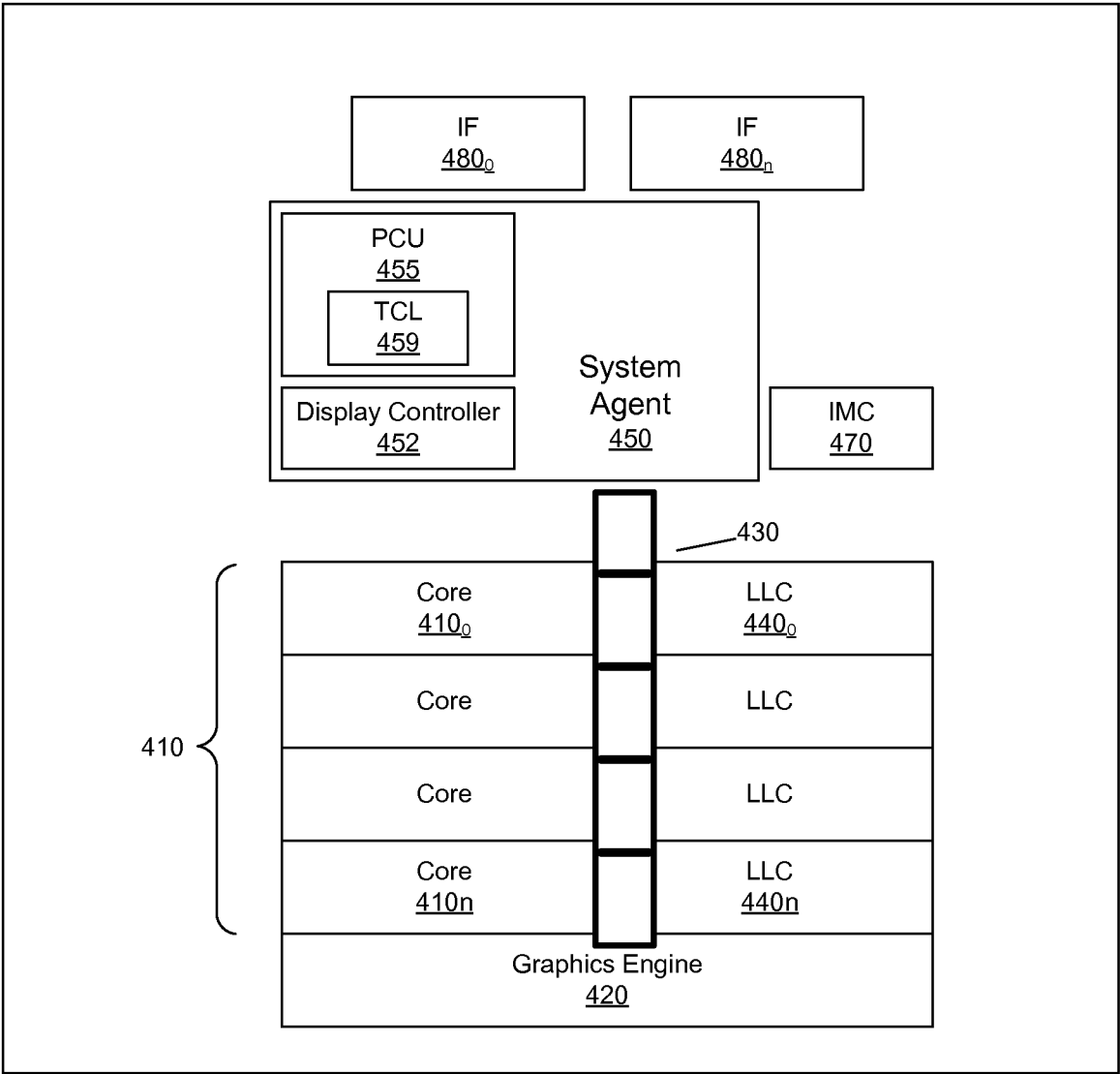


FIG. 3

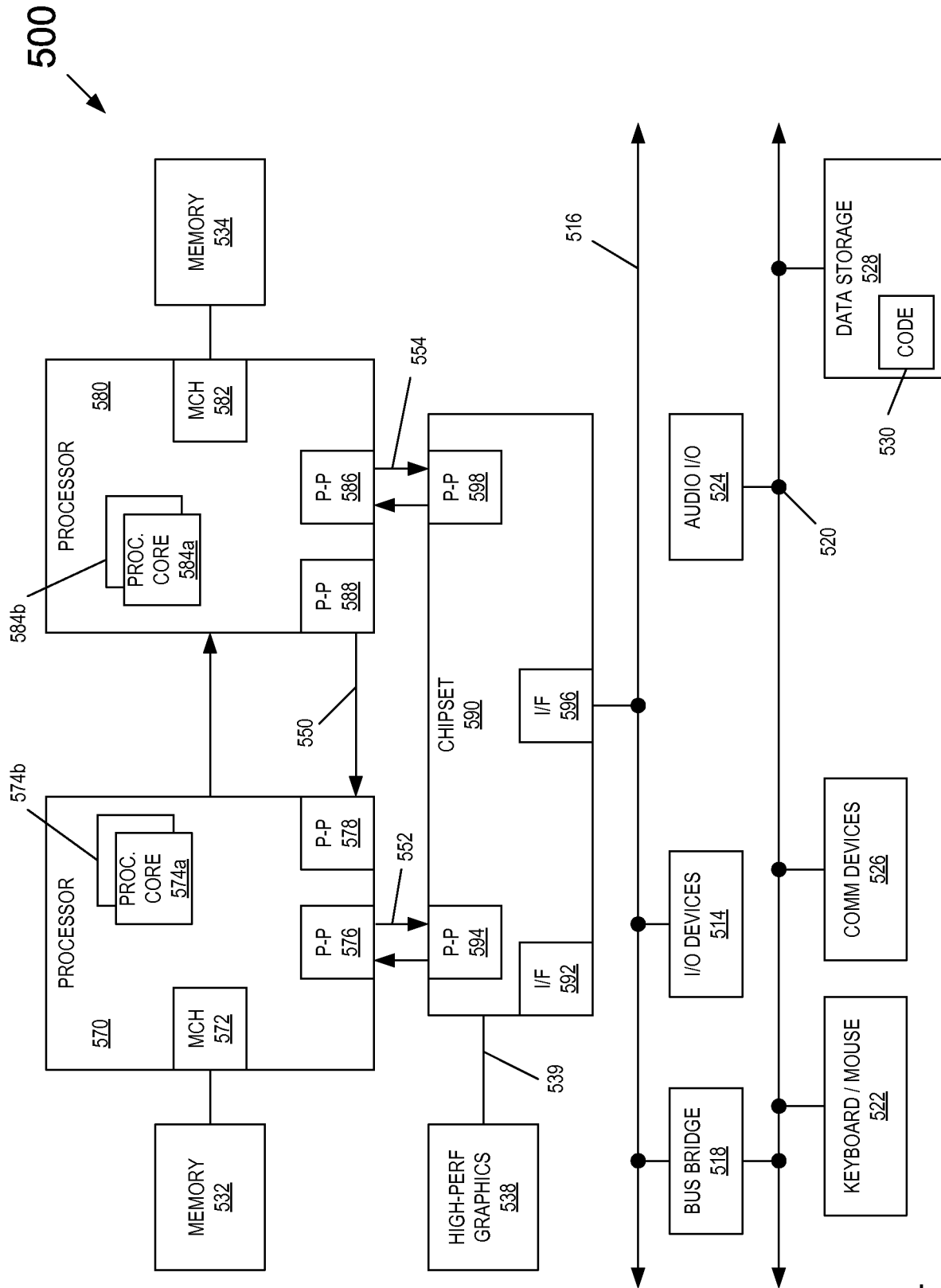


FIG. 4