(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 July 2006 (13.07.2006)

PCT

(10) International Publication Number
WO 2006/073899 A2

(51) International Patent Classification:
G06F 1/20 (2006.01)

(21) International Application Number:
PCT/US2005/046847

(22) International Filing Date:
20 December 2005 (20.12.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/026,705 30 December 2004 (30.12.2004) US

(71) Applicant *(for all designated States except US)*: **INTEL CORPORATION** [US/US]; 2200 Mission College Boulevard, Santa Clara, CA 95052 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **ROTEM, Efraim** [IL/IL]; 8 Viso St., 34400 Haifa (IL). **LAMDEN, Oren** [IL/IL]; 12 Shoshanim St., 36056 Kiryat Tivon (IL). **NAVEH, Alon** [IL/IL]; 97 Usishkin St., 47204 Ramat Hasharon (IL).

(74) Agents: **VINCENT, Lester, J.** et al.; Blakely, Sokoloff, Taylor & Zafman LLP, 12400 Wilshire Boulevard, 7th Floor, Los Angeles, CA 90025 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 2006/073899 A2

(54) Title: OPERATING POINT MANAGEMENT IN MULTI-CORE ARCHITECTURES

(57) Abstract: Systems and methods of managing operating points provide for determining the number of active cores in a plurality of processor cores. A maximum operating point is selected for at least one of the active cores based on the number of active cores. In one embodiment, the number of active cores is determined by monitoring an ACPI processor power state signal of each of the plurality of cores.

OPERATING POINT MANAGEMENT IN MULTI-CORE ARCHITECTURES

BACKGROUND

Technical Field

[0001] One or more embodiments of the present invention generally relate to operating point management. In particular, certain embodiments relate to managing operating points in multi-core processing architectures.

Discussion

[0002] The popularity of computing systems continues to grow and the demand for more complex processing architectures has experienced historical escalations. For example, multi-core processors are becoming more prevalent in the computing industry and are likely to be used in servers, desktop personal computers (PCs), notebook PCs, personal digital assistants (PDAs), wireless "smart" phones, and so on. As the number of processor cores in a system increases, the potential maximum power also increases. Increased power consumption translates into more heat, which poses a number of difficulties for computer designers and manufacturers. For example, device speed and long term reliability can deteriorate as temperature increases. If temperatures reach critically high levels, the heat can cause malfunction, degradations in lifetime or even permanent damage to parts.

[0003] While a number of cooling solutions have been developed, a gap continues to grow between the potential heat and the cooling capabilities of modern computing systems. In an effort to narrow this gap, some approaches to power management in computer processors involve the use of one or more on-die temperature sensors in conjunction with a power reduction mechanism. The power reduction mechanism is typically turned on and off (e.g., "throttled") according to the corresponding temperature sensor's state in order to reduce power consumption. Other approaches involve alternatively switching between low and high frequency/voltage operating points.

[0004] While these solutions have been acceptable under certain circumstances, there remains considerable room for improvement. For example, these solutions tend

to make the system performance more difficult to determine (i.e., the solutions tend to be "non-deterministic"). In fact, temperature based throttling is often highly dependent upon ambient conditions, which can lower the level of performance predictability. For example, on a warm day, more throttling (and therefore lower performance) is likely to occur than on a cool day for the same usage model. In addition, reducing power by throttling between operating points can add to the inconsistency of the user's experience. These drawbacks may be magnified when the gap between the dissipated power and the external cooling capabilities increases due to the presence of multiple processor cores in the system.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] The various advantages of the embodiments of the present invention will become apparent to one skilled in the art by reading the following specification and appended claims, and by referencing the following drawings, in which:

[0006] FIG. 1 is a diagram of an example of a processing architecture according to one embodiment of the invention;

[0007] FIG. 2 is a diagram of an example of a system according to one embodiment of the invention;

[0008] FIG. 3 is a flowchart of an example of a method of managing operating points according to one embodiment of the invention;

[0009] FIG. 4 is a flowchart of an example of a process of determining a number of active cores according to one embodiment of the invention; and

[0010] FIG. 5 is a flowchart of an example of a process of selecting a maximum operating point according to one embodiment of the invention.

DETAILED DESCRIPTION

[0011] In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the embodiments of the present invention. It will be evident, however, to one skilled in the art that the embodiments of the present invention may be practiced without these specific details. In other instances, specific apparatus structures and methods have not been described so

as not to obscure the embodiments of the present invention. The following description and drawings are illustrative of the embodiments of the invention and are not to be construed as limiting the embodiments of the invention.

[0012] FIG. 1 shows a processing architecture 10 having a plurality of processor cores 12 (12a, 12b), an activity module 14 and a plurality of maximum operating points 16 (16a, 16b) from which to select. The processor cores 12 can be similar to a Pentium® 4 processor core available from Intel® Corporation in Santa Clara, California, where each core 12 may be fully functional with instruction fetch units, instruction decoders, level one (L1) cache, execution units, and so on (not shown). In addition, the activity module 14 may be implemented in fixed functionality hardware such as complementary metal oxide semiconductor (CMOS) technology, in microcode, in software (e.g., as part of an operating system/OS), or any combination thereof. In the illustrated example, the activity module 14 is implemented in hardware.

[0013] In one example, each of the maximum operating points 16 includes a maximum operating frequency and voltage. The maximum operating points 16 can be determined based on knowledge of the cooling solutions available to the system and/or the thermal constraints of the system. For example, it may be determined that in a dual core architecture with only one core active, the system can be properly cooled if the active core is limited to a maximum operating frequency of 2.0 GHz (and/or a core voltage of 1.7 V). It may also be known, however, that if both cores are active, the cores should be limited to a maximum operating frequency of 1.5 GHz (and/or a core voltage of 1.35 V) in order for the cooling solution to be effective. The illustrated activity module 14 determines the number 18 of active cores in the plurality of processor cores 12 and selects a maximum operating point 17 for the active cores based on the number 18 of active cores. The maximum operating points 16 could be stored in a configuration table.

[0014] For example, the activity module 14 might make use of a configuration table such as the following Table I, to select a maximum operating point in a dual core architecture.

# Active	Max Freq.
1	2.0 GHz
2	1.5 GHz

Table I

Where the first maximum operating point 16a is assigned the value of 2.0 GHz and the second maximum operating point 16b is assigned the value of 1.5 GHz. Thus, if the activity module 14 determines that the first core 12a is active and the second core 12b is inactive, the number of active cores would be one and the first maximum operating point 16a (i.e., a maximum operating frequency of 2.0 GHz) would be selected for the first core 12a. Similarly, if it is determined that the first core 12a is inactive and the second core 12b is active, the first maximum operating point 16a (i.e., a maximum operating frequency of 2.0 GHz) would be selected for the second core 12b.

[0015] If, on the other hand, the activity module 14 determines that both the first core 12a and the second core 12b are active, the number of active cores would be two and the second maximum operating point 16b (i.e., a maximum operating frequency of 1.5 GHz) would be selected for both the first core 12a and the second core 12b. Thus, under the above scenario, the illustrated activity module 14 could determine that both cores 12a, 12b are active and therefore set the second maximum operating point 16b as the selected maximum operating point 17. Specific frequencies are given to facilitate discussion only.

[0016] By selecting the maximum operating point 17 based on the number 18 of active cores, the architecture 10 provides a number of advantages over conventional techniques. For example, the gap between the potential maximum power and the available cooling capabilities can be narrowed in a fashion that is not directly dependent upon temperature. Because the dependency on ambient temperature conditions can be minimized, more predictable performance can result. The approaches described herein are more deterministic than conventional approaches. In addition, limiting the operating point based on the number of active cores increases the effectiveness of the available cooling solutions.

[0017] The maximum operating point 17 may also be selected based on active core performance levels 19, which can be determined by the activity module 14. In particular, the processor cores 12 may be able to operate at different performance levels based on a variety of factors. For example, one approach may involve switching between low and high frequency/voltage operating points based on core utilization and/or temperature. In any case, it may be determined that an active core is running at a relatively low performance level, which may allow the other core(s) to operate at a higher performance level than would be permitted under a pure active/idle determination.

[0018] For example, it may be determined that cores 12a and 12b are active and that the first core 12a is operating at 1.0 GHz. It may also be determined that under such a condition, the second core 12b could operate at a frequency as high as 1.86 GHz without exceeding the cooling capability of the system. Rather than selecting the maximum operating point 17 for both cores to be 1.5 GHz, the activity module 14 could use the active core performance levels 19 to set a first core maximum operating point of 1.0 GHz and a second core maximum operating point of 1.86 GHz. Thus, the selected maximum operating point 17 could have a per-core component.

[0019] Turning now to FIG. 2, a system 20 having a multi-core processor 22 is shown, where the system 20 may be part of a server, desktop personal computer (PC), notebook PC, handheld computing device, etc. In the illustrated example, the processor 22 has an activity module 14', a plurality of processor cores 12' (12a'-12n') and a voltage and frequency controller 24.

[0020] The illustrated system 20 also includes one or more input/output (I/O) devices 26 and various memory subsystems coupled to the processor 22 either directly or by way of a chipset 28. In the illustrated example, the memory subsystems include a random access memory (RAM) 30 and 31 such as a fast page mode (FPM), error correcting code (ECC), extended data output (EDO) or synchronous dynamic RAM (SDRAM) type of memory, and may also be incorporated in to a single inline memory module (SIMM), dual inline memory module (DIMM), small outline DIMM (SODIMM), and so on. For example, SODIMMs have a reduced packaging height due to a slanted arrangement with respect to the adjacent circuit board. Thus, configuring the RAM 30

as a SODIMM might be particularly useful if the system 20 is part of a notebook PC in which thermal constraints are relatively tight. SODIMMs are described in greater detail in U.S. Patent No. 5,227,664 to Toshio, et al.

[0021] The memory subsystems may also include a read only memory (ROM) 32 such as a compact disk ROM (CD-ROM), magnetic disk, flash memory, etc. The illustrated RAM 30, 31 and ROM 32 include instructions 34 that may be executed by the processor 22 as one or more threads. The ROM 32 may be a basic input/output system (BIOS) flash memory. Each of the RAM 30, 31 and/or ROM 32 are able to store a configuration table 36 that can be used to select maximum operating points. The table 36, which may be calculated "on the fly" by software or pre-stored in memory, can be similar to the Table I discussed above. In this regard, the activity module 14' may include a configuration table input 38 to be used in accessing the configuration table 36.

[0022] As already discussed, the activity module 14' is able to determine the number of active cores in the plurality of processor cores 12'. The activity can be determined by monitoring a state signal 40 (40a-40n) of each of the plurality of processor cores 12' and identifying whether each state signal 40 indicates that the corresponding core is active. For example, the activity module 14' could monitor an Advanced Configuration and Power Interface (e.g., ACPI Specification, Rev. 3.0, September 2, 2004; Rev. 2.0c, August 25, 2003; Rev. 2.0, July 27, 2000, etc.) processor power state ("Cx state") signal of each of the plurality of processor cores 12'. ACPI Cx states are relatively unproblematic to monitor and therefore provide a useful solution to determining the number of active cores.

[0023] ACPI defines the power state of system processors while in the working state ("G0") as being either active (executing) or sleeping (not executing), where the power states can be applied to each processor core 12'. In particular, processor power states are designated as C0, C1, C2, C3,... Cn. The shallowest, C0, power state is an active power state where the CPU executes instructions. The C1 through Cn power states are processor sleeping states where the processor consumes less power and dissipates less heat than leaving the processor in the C0 state. While in a sleeping state, the processor core does not execute any instructions. Each processor sleeping state has a latency associated with entering and exiting the state that corresponds to the state's

power savings. In general, the longer the entry/exit latency, the greater the power savings when in the state. To conserve power, an operating system power management (OSPM) module (not shown) places the processor core into one of its supported sleeping states when idle.

[0024] The state signals 40 can also include information regarding performance levels. For example, the state signals 40 may indicate the performance level of each active core. Such a signal could be provided by ACPI performance state (Px state) signals. In particular, while in the C0 state, ACPI can allow the performance of the processor core to be altered through a defined "throttling" process and through transitions into multiple performance states (Px states). While a core is in the P0 state, it uses its maximum performance capability and may consume maximum power. While a core is in the P1 state, the performance capability of the core is limited below its maximum and consumes less than maximum power. While a core is in the Pn state, the performance capability of core is at its minimum level and consumes minimal power while remaining in an active state. State n is a maximum number and is processor or device dependent. Processor cores and devices may define support for an arbitrary number of performance states not to exceed 16 according to the ACPI Specification, Rev. 3.0.

[0025] Thus, if the illustrated activity module 14' monitors sleep state signals 40, it can identify whether each sleep state signal 40 indicates that the corresponding core is active. The activity module 14' can then search the configuration table 36 for an entry containing the number of active cores. A similar search could be conducted with respect to performance levels. Upon finding the entry, the activity module 14' may retrieve a maximum operating point, via the configuration table input 38, from the entry, where the maximum operating point enables a parameter such as frequency or core voltage to be limited.

[0026] For example, the activity module 14' can generate a limit request 42 based on the maximum operating point. As already noted, the limit request 42 may specify a maximum operating frequency and/or maximum core voltage. Thus, as the active cores submit operating point requests to the controller 24, the controller 24 ensures that none of the operating points exceed the maximum operating point specified in the limit

request 42. Simply put, the controller 24 can limit the appropriate parameter of the active cores based on the limit request 42.

[0027] Although the illustrated system 20 includes a processing architecture that contains a single package/socket, multi-core processor 22, the embodiments of the invention are not so limited. For example, a first subset of the plurality of processor cores 12 could be contained within a first processor package and a second subset of the plurality of processor cores 12 could be contained within a second processor package. Indeed, any processing architecture in which performance predictability and/or power management are issues of concern can benefit from the principles described herein. Notwithstanding, there are a number of aspects of single package/socket, multi-core processors for which the system 20 is well suited.

[0028] Turning now to FIG. 3, a method 44 of managing operating points is shown. The method 44 may be implemented in fixed functionality hardware such as complementary metal oxide semiconductor (CMOS) technology, microcode, software such as part of an operating system (OS), or any combination thereof. Processing block 46 provides for determining the number of active cores in a plurality of processor cores and/or the performance level of each of the active cores. A maximum operating point is selected for the active cores at block 48 based on the number of active cores and/or the active core performance level(s). Block 50 provides for generating a limit request based on the maximum operating point, where an operating parameter of the cores can be limited based on the limit request. The limit request may specify a maximum operating frequency and/or maximum operating voltage.

[0029] FIG. 4 shows one approach to determining the number of active cores in greater detail at block 46'. In particular, the illustrated block 52 provides for monitoring a sleep state signal of each of the plurality of processor cores. As already discussed, the sleep state signals may be ACPI Cx state signals. If the monitoring at block 52 is to include monitoring performance state data, the signals may be ACPI Px state signals. Block 54 provides for identifying whether each sleep state signal indicates that a corresponding core is active.

[0030] Turning now to FIG. 5, one approach to selecting a maximum operating point is shown in greater detail at block 48'. In the example shown, the maximum operating

point is selected based on the number of active cores. Alternatively, the selection could be based on the performance level of each active core. In particular, the illustrated block 56 provides for searching a configuration table for an entry containing the number of active cores. In one embodiment, the searching is conducted on a BIOS configuration table. The maximum operating point is retrieved from the entry at block 58. Alternatively, the maximum operating points could be calculated. Such an approach may be particularly useful if the selection of maximum operating points is based on active core performance levels. For example, the calculation could involve an averaging (weighted or unweighted) of core operating frequencies. A weighted average may be particularly useful in systems having non-symmetrical cores (i.e., large and small cores in the same system) because the larger cores could be given a greater weight due to their potentially greater contribution to the overall power consumption.

[0031] Thus, the embodiments described herein can provide for the constraining of power in multi-core processing architectures while providing predictable performance throughout most of the architecture's power range. By dynamically adjusting the maximum frequency and voltage operating point to the number of active cores in the architecture, these solutions offer a coarse-grained mechanism that can be used as a stand-alone technique or as a complement to traditional temperature-based throttling techniques.

[0032] Those skilled in the art can appreciate from the foregoing description that the broad techniques of the embodiments of the present invention can be implemented in a variety of forms. Therefore, while the embodiments of this invention have been described in connection with particular examples thereof, the true scope of the embodiments of the invention should not be so limited since other modifications will become apparent to the skilled practitioner upon a study of the drawings, specification, and following claims.

CLAIMS

What is claimed is:

1. A method comprising:

determining a number of active cores in a plurality of processor cores; and

selecting a maximum operating point for at least one of the active cores based on the number of active cores.

2. The method of claim 1, wherein the determining includes:

monitoring a sleep state signal of each of the plurality of processor cores; and

identifying whether each sleep state signal indicates that a corresponding core is active.

3. The method of claim 2, wherein the monitoring includes monitoring an Advanced Configuration and Power Interface (ACPI) processor power state (Cx state) signal.

4. The method of claim 1, wherein the selecting includes:

searching a configuration table for an entry containing the number of active cores; and

retrieving the maximum operating point from the entry.

5. The method of claim 4, wherein the searching includes searching a basic input/output system (BIOS) configuration table.

6. The method of claim 1, further including generating a limit request based on the maximum operating point.

7. The method of claim 6, wherein the generating includes generating a limit request that specifies a maximum operating frequency.

8. The method of claim 7, wherein generating the limit request further includes generating a limit request that specifies a maximum core voltage.

9. The method of claim 6, further including limiting an operating parameter of the active cores based on the limit request.

10. The method of claim 1, further including determining a performance level of each of the active cores, the selecting being further based on the performance levels.

11. The method of claim 10, wherein determining each performance level includes monitoring an Advanced Configuration and Power Interface (ACPI) performance state (Px state) signal of a corresponding core.

12. An apparatus comprising:

a plurality of processor cores; and

an activity module to determine a number of active cores in the plurality of processor cores and select a maximum operating point for at least one of the active cores based on the number of active cores.

13. The apparatus of claim 12, wherein the activity module is to monitor a sleep state signal of each of the plurality of processor cores and identify whether each sleep state signal indicates that a corresponding core is active.

14. The apparatus of claim 13, wherein the activity module is to monitor an Advanced Configuration and Power Interface (ACPI) processor power state (Cx state) signal.

15. The apparatus of claim 12, wherein the activity module is to search a configuration table for an entry containing the number of active cores and retrieve the maximum operating point from the entry.

16. The apparatus of claim 15, wherein the activity module is to search a basic input/output system (BIOS) configuration table to obtain the entry.

17. The apparatus of claim 12, wherein the activity module is to generate a limit request based on the maximum operating point.

18. The apparatus of claim 17, wherein the limit request is to specify a maximum operating frequency.

19. The apparatus of claim 18, wherein the limit request is to further specify a maximum core voltage.

20. The apparatus of claim 17, further including a controller to limit an operating parameter of the active cores based on the limit request.

21. The apparatus of claim 12, further including a processor package that contains the plurality of processor cores.

22. The apparatus of claim 12, further including:

a first processor package that contains a first subset of the plurality of processor cores; and

a second processor package that contains a second subset of the plurality of processor cores.

23. A system comprising:

a small outline dual inline memory module (SODIMM); and

a processing architecture coupled to the SODIMM, the architecture including a plurality of processor cores and an activity module, the activity module to determine a number of active cores in the plurality of processor cores and select a maximum operating point for at least one of the active cores based on the number of active cores.

24. The system of claim 23, wherein the activity module is to monitor a sleep state signal of each of the plurality of processor cores and identify whether each sleep state signal indicates that a corresponding core is active.

25. The system of claim 23, further including a memory to store a configuration table, the activity module to search the configuration table for an entry containing the number of active cores and retrieve the maximum operating point from the entry.

26. The system of claim 23, further including a controller, the activity module to generate a limit request based on the maximum operating point, the controller to limit an operating parameter of the active cores based on the limit request.

27. A method comprising:

monitoring an Advanced Configuration and Power Interface (ACPI) processor power state (Cx state) signal of each of a plurality of processor cores;

identifying whether each Cx state signal indicates that a corresponding core is active to determine a number of active cores in the plurality of processor cores;

searching a configuration table for an entry containing the number of active cores;

retrieving a maximum operating point from the entry;

generating a limit request based on the maximum operating point; and

limiting an operating parameter of the active cores based on the limit request.

28. The method of claim 27, wherein the generating includes generating a limit request that specifies a maximum operating frequency.

29. The method of claim 28, wherein generating the limit request further includes generating a limit request that specifies a maximum core voltage.

FIG. 1

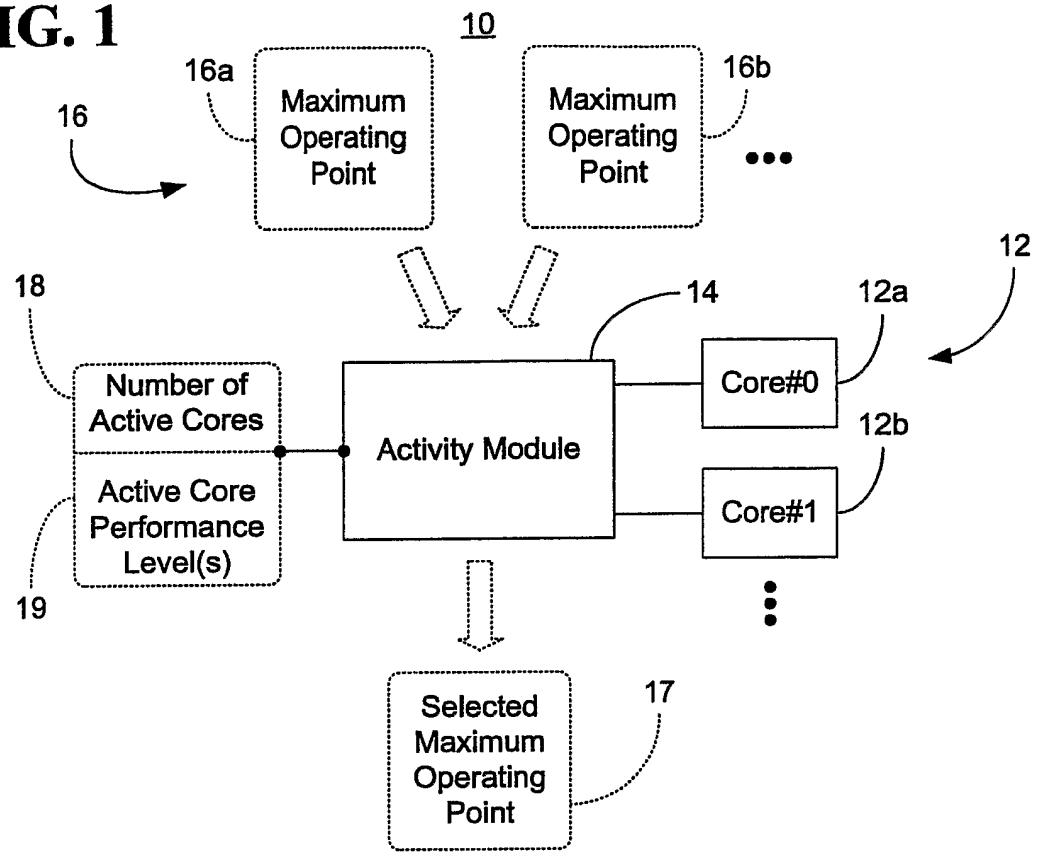

10
16
16a
Maximum Operating Point
Maximum Operating Point
16b
12
18
14
12a
Number of Active Cores
Active Core Performance Level(s)
19
Activity Module
Core#0
12b
Core#1
Selected Maximum Operating Point
17

FIG. 3

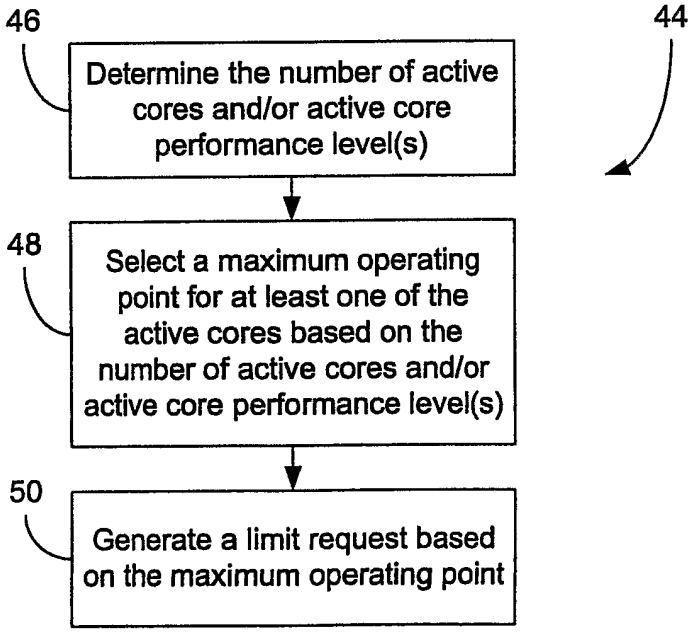

46
Determine the number of active cores and/or active core performance level(s)
44
48
Select a maximum operating point for at least one of the active cores based on the number of active cores and/or active core performance level(s)
50
Generate a limit request based on the maximum operating point

FIG. 2

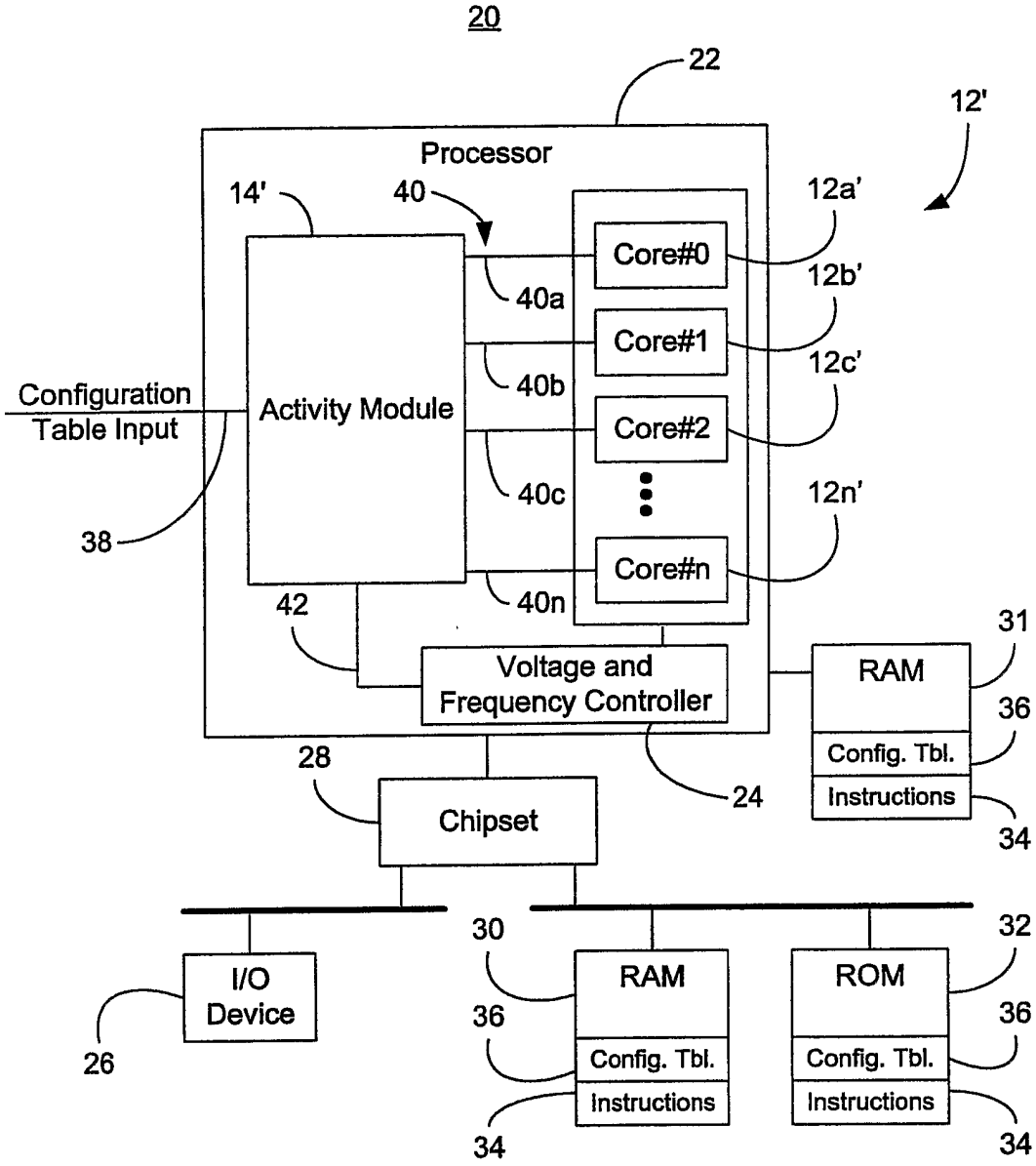

20
22
12'
Processor
14'
40
12a'
Core#0
40a
12b'
Core#1
40b
12c'
Core#2
40c
12n'
Core#n
40n
Configuration
Table Input
38
Activity Module
42
Voltage and
Frequency Controller
24
RAM
31
36
Config. Tbl.
Instructions
34
28
Chipset
I/O
Device
26
30
RAM
36
Config. Tbl.
Instructions
34
32
ROM
36
Config. Tbl.
Instructions
34

FIG. 4

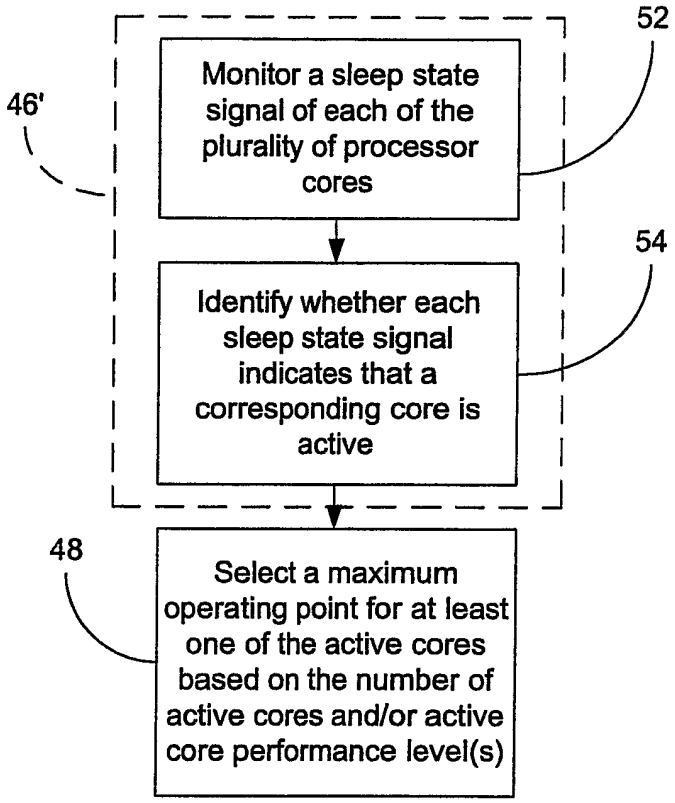
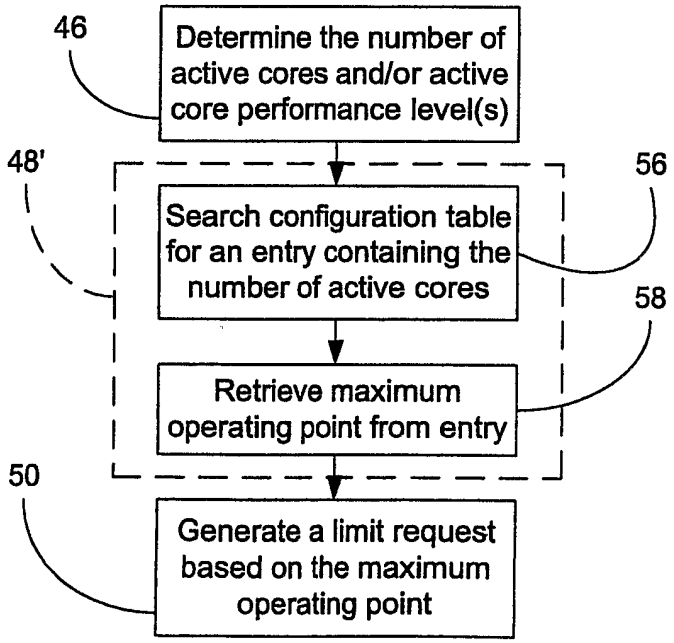

46'
52
Monitor a sleep state signal of each of the plurality of processor cores
54
Identify whether each sleep state signal indicates that a corresponding core is active
48
Select a maximum operating point for at least one of the active cores based on the number of active cores and/or active core performance level(s)

FIG. 5


46
Determine the number of active cores and/or active core performance level(s)
48'
56
Search configuration table for an entry containing the number of active cores
58
Retrieve maximum operating point from entry
50
Generate a limit request based on the maximum operating point