

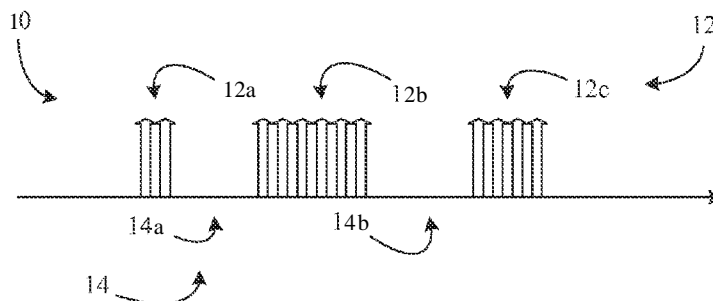


- (51) **International Patent Classification:**  
*G06F 1/32* (2006.01)
- (21) **International Application Number:**  
PCT/US2012/066656
- (22) **International Filing Date:**  
27 November 2012 (27.11.2012)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**  
3433/DEL/201 1 29 November 2011 (29.11.2011) IN
- (71) **Applicant (for all designated States except US):** **INTEL CORPORATION** [US/US]; 2200 Mission College Boulevard, Santa Clara, California 95054 (US).
- (72) **Inventors; and**
- (73) **Applicants (for US only):** **WANG, Ren** [US/US]; 9137 NW Esson Ct., Portland, Oregon 97229 (US). **TAI, Tsung-Yuan C.** [US/US]; 12709 NW Majestic Sequoia Way, Portland, Oregon 97229 (US).  **TSAI, JR-Shian** [US/US]; 6657 NW 165th Ave., Portland, Oregon 97229 (US). **FLEMING, Bruce L.** [US/US]; 1481 Morning Star Dr., Morgan Hill, California 95037 (US). **MURALIDHAR, Rajeev D.** [IN/IN]; 305 6th Block, 2nd Phase, BSK 3rd Stage, Bangalore, KA 560085 (IN). **ERGIN, Mesut A.** [TR/US]; 15270 NW Nightshade Drive, Portland, Oregon 97229 (US). **IYER, Prakash N.** [US/US]; 12535 NW Lilywood Drive, Portland, Oregon 97229 (US). **SESHADRI, Harinarayanan** [IN/IN]; 112, Haritha Apartments, 301, 11 cross Malleswaram, Bangalore, KA 560003 (IN).
- (74) **Agents:** **JORDAN, B. Delano** et al; Jordan IP Law, PC, c/o CPA GLOBAL, P.O. Box 52050, Minneapolis, Minnesota 55402 (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).

**Published:**

— with international search report (Art. 21(3))

(54) **Title:** DYNAMICALLY ENTERING LOW POWER STATES DURING ACTIVE WORKLOADS



**FIG. 1**

(57) **Abstract:** Systems and methods may provide for identifying runtime information associated with an active workload of a platform, and making an active idle state determination for the platform based on at least in part the runtime information. In addition, a low power state of a shared resource on the platform may be controlled concurrently with an execution of the active workload based on at least in part the active idle state determination.



## DYNAMICALLY ENTERING LOW POWER STATES DURING ACTIVE WORKLOADS

### CROSS-REFERENCE TO RELATED APPLICATIONS

The present application is related to U.S. Patent Application No. 12/888,855, filed  
5 on September 23, 2010, and U.S. Patent Application No. 12/644,720, filed on December  
22, 2009.

### BACKGROUND

#### Technical Field

Embodiments generally relate to power management in computing platforms.  
10 More particularly, embodiments relate to the use of low power states during active  
workloads.

#### Discussion

In conventional mobile computing platforms, low power states may be used to  
reduce power consumption and extend battery life. The usage of low power states may be  
15 limited, however, to time periods when the platform is idle. Such an approach may not  
achieve optimal energy efficiency and performance in certain circumstances.

### BRIEF DESCRIPTION OF THE DRAWINGS

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  
20 claims, and by referencing the following drawings, in which:

FIG. 1 is a timeline of an example of an active workload having active idle  
windows according to an embodiment;

FIG. 2 is a block diagram of an example of dynamic power management logic  
according to an embodiment;

25 FIGS. 3A-3C are block diagrams of examples of system power hierarchies  
according to embodiments;

FIG. 4 is a block diagram of an example of dynamic power management logic  
implemented in firmware according to an embodiment;

FIG. 5 is a flowchart of an example of a method of implementing active idle states  
30 in a platform according to an embodiment; and

FIG. 6 is a flowchart of an example of a method of making active idle state  
determinations and controlling low power states according to an embodiment.

## DETAILED DESCRIPTION

Turning now to FIG. 1, a timeline of an active workload 10 for a platform is shown in which a plurality of active windows 12 (12a-12c) are interspersed with a plurality of "active idle" windows 14 (14a, 14b). The active workload 10 could correspond to a computing activity (e.g., web browsing, video playback, etc.) that utilizes far less than 100% of the processing resources of the platform. Thus, while the platform may perform tasks, execute operations, process events, and so forth, during the active windows 12, at least a portion of the platform may be inactive/idle during the active idle windows 14. As will be discussed in greater detail, rather than maintaining the platform in a relatively high power state during execution of the active workload 10, a strategy may be employed in which the platform is placed in an "active idle state" to optimize platform power consumption during execution of the active workload 10.

FIG. 2 shows a block diagram of a platform 16 having software (e.g., operating system/OS, applications) 20, shared resources 22 (22a-22c), and devices 24. The platform 16 may be part of a mobile platform having computing functionality (e.g., personal digital assistant/PDA, laptop, smart tablet), communications functionality (e.g., wireless smart phone), imaging functionality, media playing functionality (e.g., smart television/TV), or any combination thereof (e.g., mobile Internet device/MID). The shared resources 22 may therefore include, for example, one or more processors such as a CPU (central processing unit) 22a, one or more supporting components 22b such as a memory structure and/or chipset, and one or more host controllers 22c such as a USB (Universal Serial Bus, e.g., USB Specification 3.0, USB Implementers Forum) controller. In addition, the devices 24 may include storage (e.g., flash drives, hard disk drives, solid state disks), network controllers, audio input/output (IO) devices, peripheral devices, and so forth. Thus, the devices 24 might interact with the platform 16 via the shared resources 22, and could be considered downstream components with respect to the shared resources 22 to the extent that a given device 24 has a power dependency on a particular shared resource 22.

The illustrated platform 16 also includes dynamic power management logic (DPML) 18 that obtains runtime information from the software 20, shared resources 22 and devices 24, and makes active idle state determinations for the platform 16 based on the runtime information. The runtime information, which can be associated with an active workload of the platform 16, may include, for example, idle duration information, latency tolerance requirement (LTR) information, resource requirement information, and so forth.

As will be discussed in greater detail, the runtime information can enable the logic 18 to determine whether to place one or more of the shared resources 22 and/or devices 24 in a low power state during execution of the active workload. The low power state could be an ACPI (Advanced Configuration and Power Interface, e.g., ACPI Specification, Rev. 4.0a, April 5, 2010) state or other appropriate low power state. While the logic 18 is shown as residing on a separate component 26, the logic 18 could be implemented elsewhere on the platform 16 such as in the software 20, in the shared resources 22, in the devices 24, or any combination thereof.

Turning now to FIG. 3A a system power hierarchy 28 is shown in which different nodes of a system have different power dependencies. In the illustrated example, a shared resource "SI" has the devices "D1" and "D2" (e.g., leaf nodes) as downstream components, and the shared resource SI is itself a downstream component of another shared resource "S2" (along with another device "D3"). Thus, the decision of whether to place the shared resource SI in a low power state may depend at least in part on the status of the devices D1 and D2. Similarly, the decision of whether to place the shared resource S2 in a low power state may depend at least in part on the status of the shared resource S1 and the device D3, in the example shown. In particular, the leaf nodes may autonomously manage their own power states according to workloads, and might be able to enter certain low power states by themselves when long enough idle durations such as active idle windows 14 (FIG. 1) are detected. For example, when a user is watching local video playback without network activity, a network controller of the platform could enter a low power state, and a CPU of the platform could enter a low power state when it is idle. In turn, if all nodes depending from a given shared resource are in a low power state, the shared resource in question can potentially enter a lower power state as well.

FIG. 3B demonstrates a scenario in which a system power hierarchy 30 includes a set of low power nodes 32. In the illustrated example, the shared resource SI has been placed in a low power state because its downstream component devices D1 and D2 are also in a low power state. Similarly, a shared resource node "S5" has been placed in a low power state because its downstream component devices "D7" and "D8" are also in a low power state. A shared resource "S3", on the other hand, is maintained in an active state because one of its downstream component devices, "D5", is not in a low power state.

The decision of whether to place a shared resource or device in a low power state may also depend, however, on other runtime factors. For example, a shared resource or

device might not be able to enter a low power state if a projected active idle window is shorter than the energy break-even time for the low power state in question. In this regard, there may be transition-related power consumption overhead as well as latency overhead that could offset the advantages associated with the reduced power consumption of a low power state if the active idle window is not long enough. Thus, the energy break-even time can generally indicate the amount of time the component would have to remain idle in order to justify transitioning to and from the low power state under consideration.

Latency tolerance requirements may also be a consideration. For example, if the resume latency of a low power state is longer than the latency tolerance requirement (LTR) associated with a certain workload, the component could be maintained in the active state in order to satisfy the LTR. In addition, some workloads can have specific resource requirements, wherein if a shared resource is included in the resource requirement for the workload, then that shared resource may be prevented from entering a low power state during execution of that workload.

Thus, FIG. 3C shows a system power hierarchy 34 in which a set of low power nodes 36 includes the downstream component devices D7 and D8, but does not include the shared resource S5. In the illustrated example, the shared resource S5 might have been maintained in the active state because, for example, the active idle window does not exceed the energy break-even time of a low power state of the shared resource S5, the latency of the low power state exceeds a latency tolerance requirement of the active workload, a resource requirement of the active workload includes the shared resource S5, and so forth.

FIG. 4 shows a platform 38 in which dynamic power management logic such as the DPML 18 (FIG. 2) is implemented in firmware 40. In the illustrated example, runtime information 42 is obtained from an operating system (OS) 44, wherein the runtime information 42 might include Quality of Service (QoS) information such as monitor-wait (mwait) instructions, idle duration information such as "time to next timer event" (TNTE) information, and so forth. The mwait instructions from the OS 44 could imply software latency requirements, and the TNTE information might also be used as a "hint" to determine the next idle duration. The TNTE information can either be communicated from the OS 44 to the firmware 40, or accessed via an OS timer. The illustrated firmware 40 also receives runtime information 46 from one or more platform devices/subsystems 48, which could include device drivers, and so forth. Idle duration information may also

be obtained from an interrupt handler unit (not shown) residing in the firmware 40, and/or elsewhere in the platform 38.

The illustrated firmware 40 uses the runtime information 42, 46 to make active idle state determinations for the platform 38, and to control low power states of one or more shared resources 50 of the platform 38 during execution of active workloads. In particular, the firmware 40 may conduct fine grained power management by dynamically placing one or more of the shared resources 50 into active idle states based at least in part on the active idle state determinations. In addition, one or more of the devices 48 may be downstream components with respect to the shared resources 50. Accordingly, the active idle state determinations may take into consideration the runtime information 42, 46 in order to ensure optimal operation of the platform 38 with regard to latency, break-even times, resource requirements, etc. The firmware 40 may also include a break-event module 52 to coalesce and/or align break events in order to create active idle windows that may be used to more efficiently manage platform power. Implementing the illustrated dynamic power management logic in the firmware 40 may minimize modifications to the OS 44, enable the component with the best knowledge of the platform status to be leveraged in active idle state determinations, and reduce operation latency by using a component a component that is closer to hardware than the OS kernel. Other system implementations, however, may also be used.

Turning now to FIG. 5, a method 54 of implementing active idle states is shown. The method 54 may be implemented as a set of logic instructions stored in a machine- or computer-readable storage medium such as random access memory (RAM), read only memory (ROM), programmable ROM (PROM), firmware, flash memory, etc., in configurable logic such as programmable logic arrays (PLAs), field programmable gate arrays (FPGAs), complex programmable logic devices (CPLDs), in fixed-functionality logic hardware using circuit technology such as application specific integrated circuit (ASIC), complementary metal oxide semiconductor (CMOS) or transistor-transistor logic (TTL) technology, or any combination thereof.

Illustrated processing block 56 provides for identifying runtime information associated with an active workload of a platform. An active idle state determination may be made for the platform at block 58 based on at least in part the runtime information. In addition, block 60 provides for controlling a low power state of one or more shared

resources on the platform during execution of the active workload based on at least in part the active idle state determination.

FIG. 6 shows one approach to making active idle state determinations and controlling low power states in greater detail. Generally, a method 62 might be conducted for a low power state of a shared resource during execution of an active load. In particular, illustrated block 64 provides for determining whether all of the downstream components of the shared resource are idle. If not, illustrated block 66 keeps the shared resource in the active state. Otherwise, a determination may be made at block 68 as to whether an idle duration (e.g., active idle window) associated with the active workload exceeds an energy break-even point of the low power state. As already noted, energy break-even points may be deployed in order to ensure that low power state transitions take into account latency and power consumption overhead.

Illustrated block 70 confirms that the resume latency of the low power state does not exceed any latency tolerance requirements of the active workload, while block 72 may confirm that the shared resource is not included in any resource requirements associated with the active workload. If the conditions in the illustrated blocks 64, 68, 70 and 72 are satisfied, block 74 may place the shared resource in the low power state, wherein the method 62 may be conducted for each shared resource of a platform. Moreover, a given shared resource may have multiple potential low power states, wherein the method 62 could be conducted for each low power state of the shared resource.

Techniques described herein may therefore introduce dynamic platform low power states, e.g., "active idle states", that enable optimal platform power management during active workloads. In particular, dynamic power management logic may determine how the platform should enter active idle states to dynamically maximize platform power savings. Such an approach can enable longer battery life under usage models that traditionally have limited ability to conserve power.

Embodiments may therefore provide for a computer implemented method in which runtime information associated with an active workload of a platform is identified. The method may also involve making an active idle state determination for the platform based on at least in part the runtime information. In addition, a low power state of a shared resource on the platform may be controlled concurrently with an execution of the active workload based on at least in part the active idle state determination.

Embodiments may also include a non-transitory computer readable storage medium having a set of instructions which, if executed by a processor, cause a platform to identify runtime information associated with an active workload of the platform. The instructions can also cause the platform to make an active idle state determination for the platform based on at least in part the runtime information, and control a low power state of a shared resource on the platform concurrently with an execution of the active workload based on at least in part the active idle state determination.

In addition, embodiments can include an apparatus having logic to identify runtime information associated with an active workload of a platform, and make an active idle state determination for the platform based on at least in part the runtime information. The logic may also control a low power state of a shared resource on the platform concurrently with an execution of the active workload based on at least in part the runtime information.

Other embodiments may include a platform having a shared resource and logic to identify runtime information associated with an active workload of the platform. The logic can also make an active idle state determination for the platform based on at least in part the runtime information, and control a low power state of the shared resource concurrently with an execution of the active workload based on at least in part the active idle state determination.

Embodiments of the present invention are applicable for use with all types of semiconductor integrated circuit ("IC") chips. Examples of these IC chips include but are not limited to processors, controllers, chipset components, programmable logic arrays (PLAs), memory chips, network chips, systems on chip (SoCs), SSD/NAND controller ASICs, and the like. In addition, in some of the drawings, signal conductor lines are represented with lines. Some may be different, to indicate more constituent signal paths, have a number label, to indicate a number of constituent signal paths, and/or have arrows at one or more ends, to indicate primary information flow direction. This, however, should not be construed in a limiting manner. Rather, such added detail may be used in connection with one or more exemplary embodiments to facilitate easier understanding of a circuit. Any represented signal lines, whether or not having additional information, may actually comprise one or more signals that may travel in multiple directions and may be implemented with any suitable type of signal scheme, e.g., digital or analog lines implemented with differential pairs, optical fiber lines, and/or single-ended lines.

Example sizes/models/values/ranges may have been given, although embodiments of the present invention are not limited to the same. As manufacturing techniques (e.g., photolithography) mature over time, it is expected that devices of smaller size could be manufactured. In addition, well known power/ground connections to IC chips and other components may or may not be shown within the figures, for simplicity of illustration and discussion, and so as not to obscure certain aspects of the embodiments of the invention. Further, arrangements may be shown in block diagram form in order to avoid obscuring embodiments of the invention, and also in view of the fact that specifics with respect to implementation of such block diagram arrangements are highly dependent upon the platform within which the embodiment is to be implemented, i.e., such specifics should be well within purview of one skilled in the art. Where specific details (e.g., circuits) are set forth in order to describe example embodiments of the invention, it should be apparent to one skilled in the art that embodiments of the invention can be practiced without, or with variation of, these specific details. The description is thus to be regarded as illustrative instead of limiting.

The term "coupled" may be used herein to refer to any type of relationship, direct or indirect, between the components in question, and may apply to electrical, mechanical, fluid, optical, electromagnetic, electromechanical or other connections. In addition, the terms "first", "second", etc. may be used herein only to facilitate discussion, and carry no particular temporal or chronological significance unless otherwise indicated.

Those skilled in the art will 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

We claim:

1. A computer implemented method comprising:  
identifying runtime information associated with an active workload of a platform;  
5 making an active idle state determination for the platform based on at least in part  
the runtime information; and  
controlling a low power state of a shared resource on the platform concurrently  
with an execution of the active workload based on at least in part the active idle state  
determination.
- 10 2. The method of claim 1, wherein the runtime information includes at least  
one of an idle duration associated with the active workload, a latency tolerance  
requirement associated with the active workload, and a resource requirement associated  
with the active workload.
- 15 3. The method of claim 2, further including placing the shared resource in the  
low power state if,  
one or more downstream components of the shared resource are idle, and  
the idle duration associated with the active workload exceeds an energy break-even  
time associated with the low power state.
- 20 4. The method of claim 3, further including accessing an operating system  
timer to identify the idle duration.
5. The method of claim 3, further including receiving the idle duration from  
an interrupt handler unit.
- 25 6. The method of claim 2, further including placing the shared resource in the  
low power state if,  
one or more downstream components of the shared resource are idle, and  
a latency associated with the low power state does not exceed the latency tolerance  
requirement.
- 30 7. The method of claim 2, further including placing the shared resource in the  
low power state if,  
one or more downstream components of the shared resource are idle, and  
the resource requirement does not include the shared resource.
8. A non-transitory computer readable storage medium comprising a set of  
instructions which, if executed by a processor, cause a platform to:

identify runtime information associated with an active workload of the platform;  
make an active idle state determination for the platform based on at least in part the  
runtime information; and

control a low power state of a shared resource on the platform concurrently with an  
5 execution of the active workload based on at least in part the active idle state  
determination.

9. The medium of claim 8, wherein the runtime information is to include at  
least one of an idle duration associated with the active workload, a latency tolerance  
requirement associated with the active workload, and a resource requirement associated  
10 with the active workload.

10. The medium of claim 9, wherein the instructions, if executed, cause the  
platform to place the shared resource in the low power state if,  
one or more downstream components of the shared resource are idle, and  
the idle duration associated with the active workload exceeds an energy  
15 break-even time associated with the low power state.

11. The medium of claim 10, wherein the instructions, if executed, cause the  
platform to access an operating system timer to identify the idle duration.

12. The medium of claim 10, wherein the instructions, if executed, cause the  
platform to receive the idle duration from an interrupt handler unit.

20 13. The medium of claim 9, wherein the instructions, if executed, cause the  
platform to place the shared resource in the low power state if,  
one or more downstream components of the shared resource are idle, and  
a latency associated with the low power state does not exceed the latency  
tolerance requirement.

25 14. The medium of claim 9, wherein the instructions, if executed, cause the  
platform to place the shared resource in the low power state if,  
one or more downstream components of the shared resource are idle, and  
the resource requirement does not include the shared resource.

15. An apparatus comprising:  
30 logic to,  
identify runtime information associated with an active workload of a  
platform,

make an active idle state determination for the platform based on at least in part the runtime information, and

control a low power state of a shared resource on the platform concurrently with an execution of the active workload based on at least in part the active idle state determination.

16. The apparatus of claim 15, wherein the runtime information is to include at least one of an idle duration associated with the active workload, a latency tolerance requirement associated with the active workload, and a resource requirement associated with the active workload.

17. The apparatus of claim 16, wherein the logic is to place the shared resource in the low power state if,

one or more downstream components of the shared resource are idle, and the idle duration associated with the active workload exceeds an energy break-even time associated with the low power state.

18. The apparatus of claim 17, wherein the logic is to access an operating system timer to identify the idle duration.

19. The apparatus of claim 17, wherein the logic is to receive the idle duration from an interrupt handler unit.

20. The apparatus of claim 16, wherein the logic is to place the shared resource in the low power state if,

one or more downstream components of the shared resource are idle, and a latency associated with the low power state does not exceed the latency tolerance requirement.

21. The apparatus of claim 16, wherein the logic is to place the shared resource in the low power state if,

one or more downstream components of the shared resource are idle, and the resource requirement does not include the shared resource.

22. A platform comprising:

a shared resource; and

logic to,

identify runtime information associated with an active workload of the platform,

make an active idle state determination for the platform based on at least in part the runtime information, and

control a low power state of the shared resource concurrently with an execution of the active workload based on at least in part the active idle state determination.

23. The platform of claim 22, wherein the runtime information is to include at least one of an idle duration associated with the active workload, a latency tolerance requirement associated with the active workload, and a resource requirement associated with the active workload.

24. The platform of claim 23, further including one or more downstream components of the shared resource, wherein the logic is to place the shared resource in the low power state if,

at least one of the one or more downstream components is idle, and the idle duration associated with the active workload exceeds an energy

break-even time associated with the low power state.

25. The platform of claim 24, wherein the logic is to access an operating system timer to identify the idle duration.

26. The platform of claim 24, further including an interrupt handler unit, wherein the logic is to receive the idle duration from the interrupt handler unit.

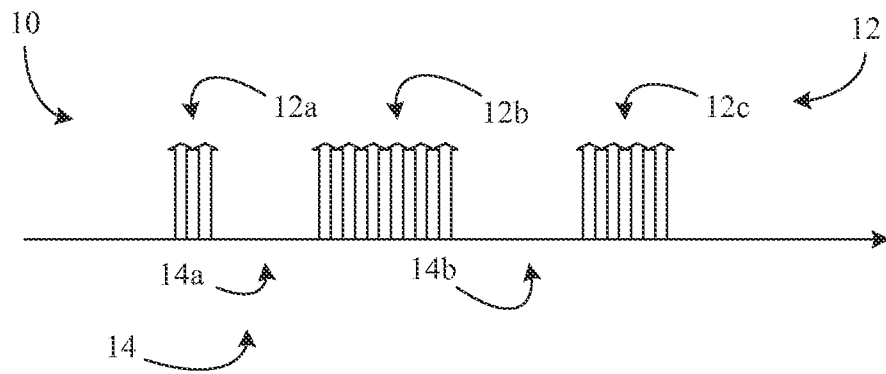
27. The platform of claim 23, further including one or more downstream components of the shared resource, wherein the logic is to place the shared resource in the low power state if,

at least one of the one or more downstream components is idle, and a latency associated with the low power state does not exceed the latency

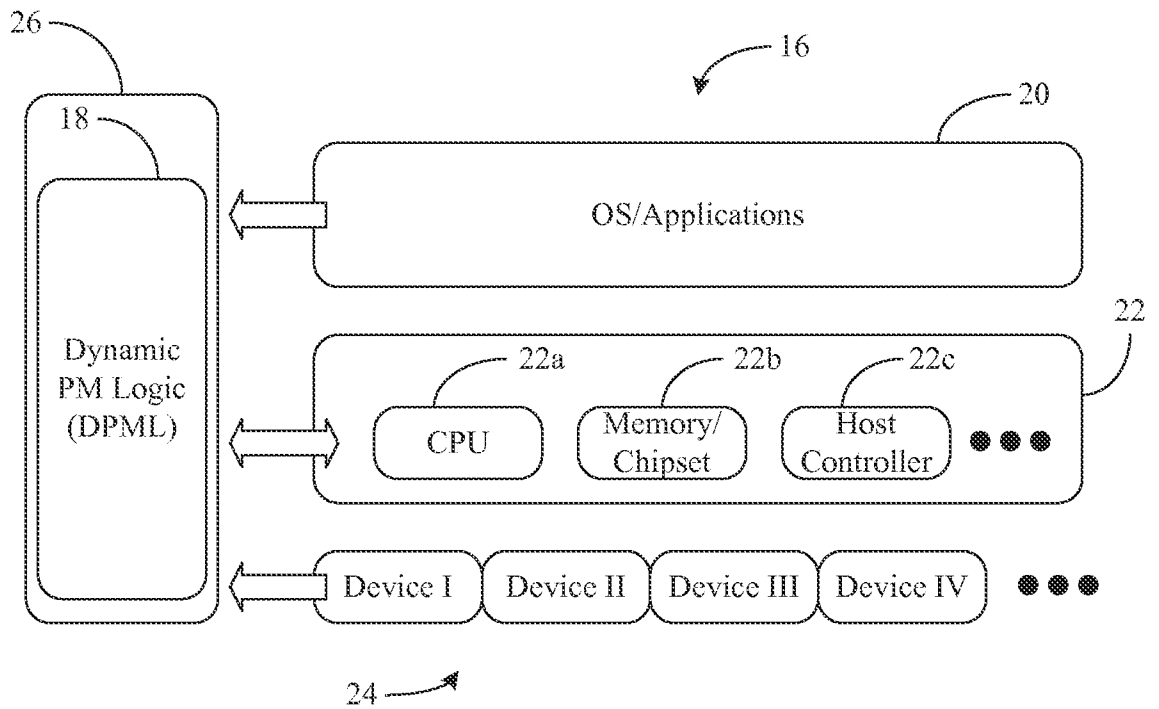
tolerance requirement.

28. The platform of claim 23, further including one or more downstream components of the shared resource, wherein the logic is to place the shared resource in the low power state if,

at least one of the one or more downstream components is idle, and the resource requirement does not include the shared resource.



**FIG. 1**



**FIG. 2**

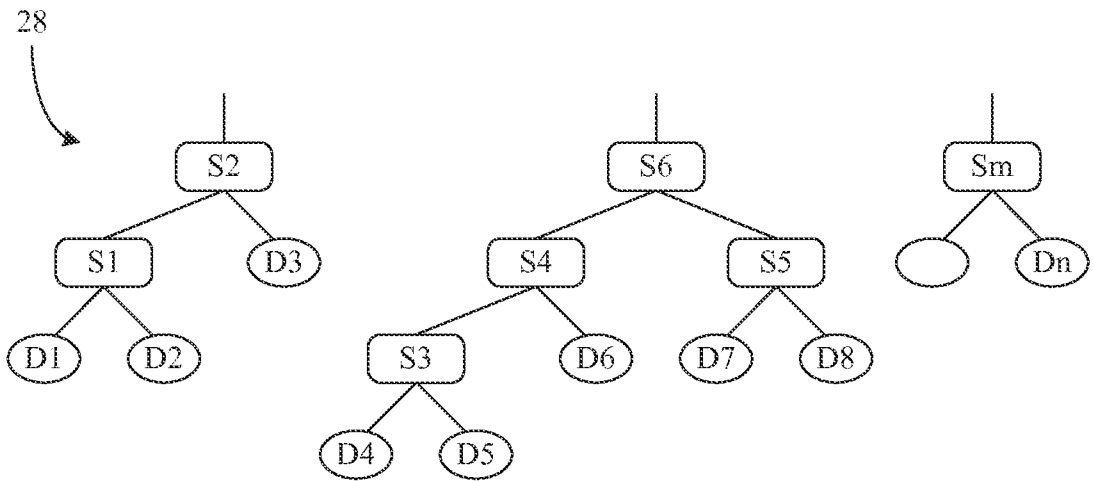


FIG. 3A

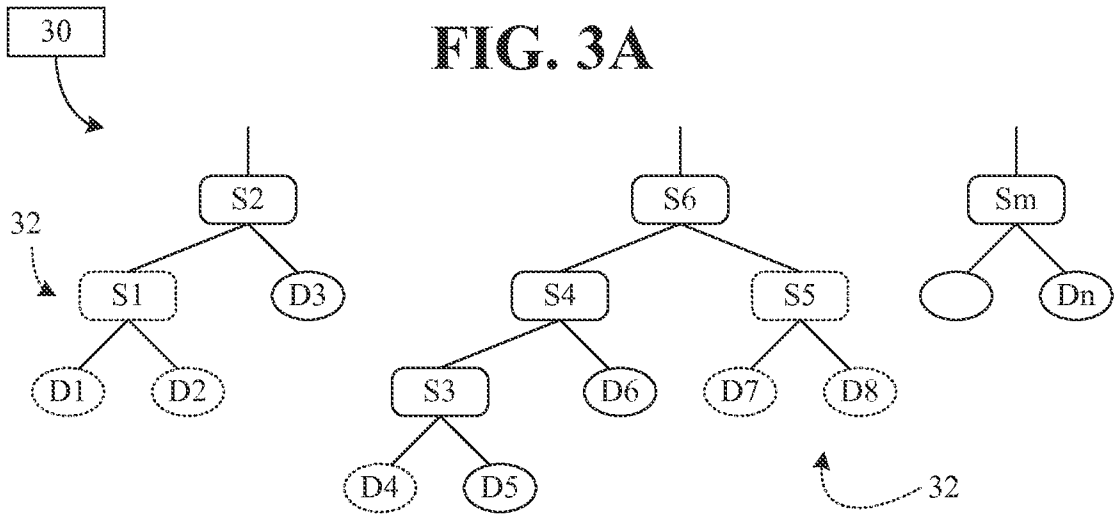


FIG. 3B

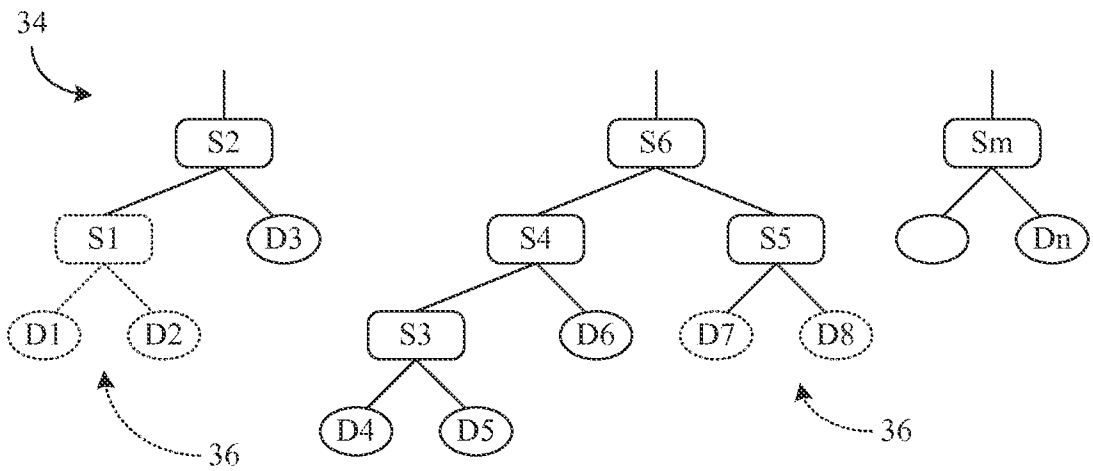


FIG. 3C

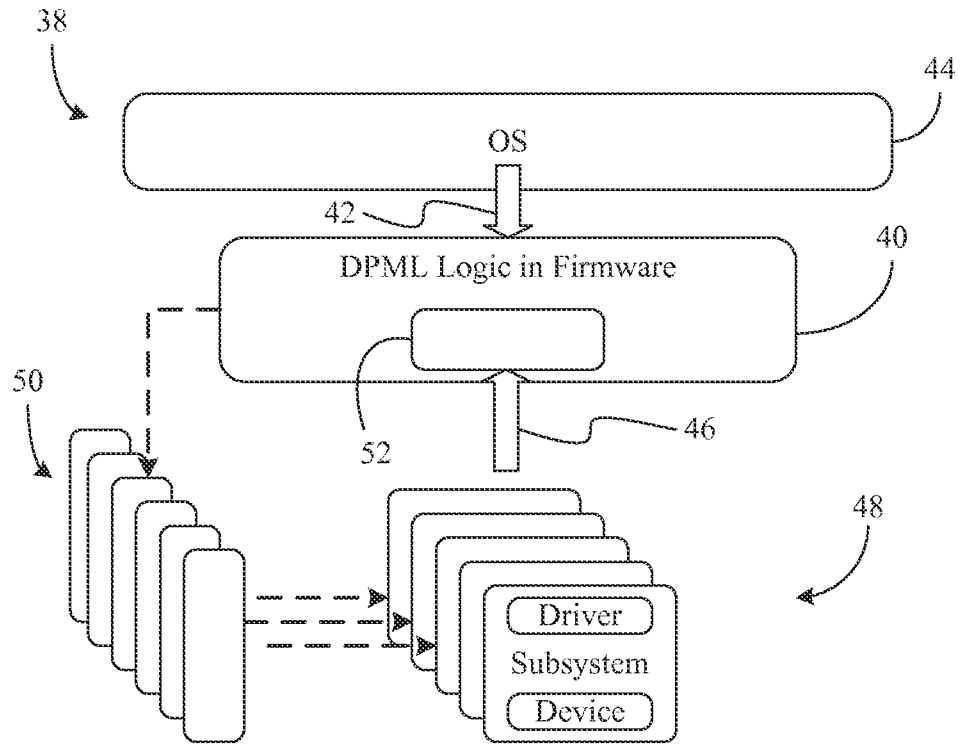


FIG. 4

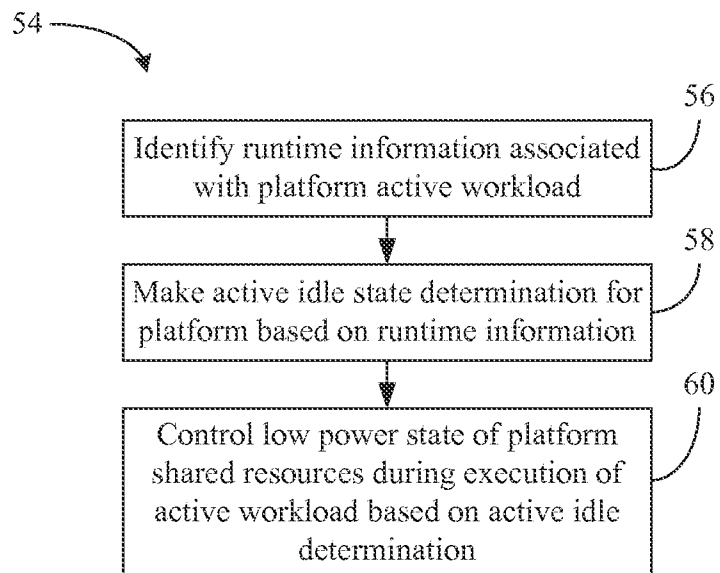


FIG. 5

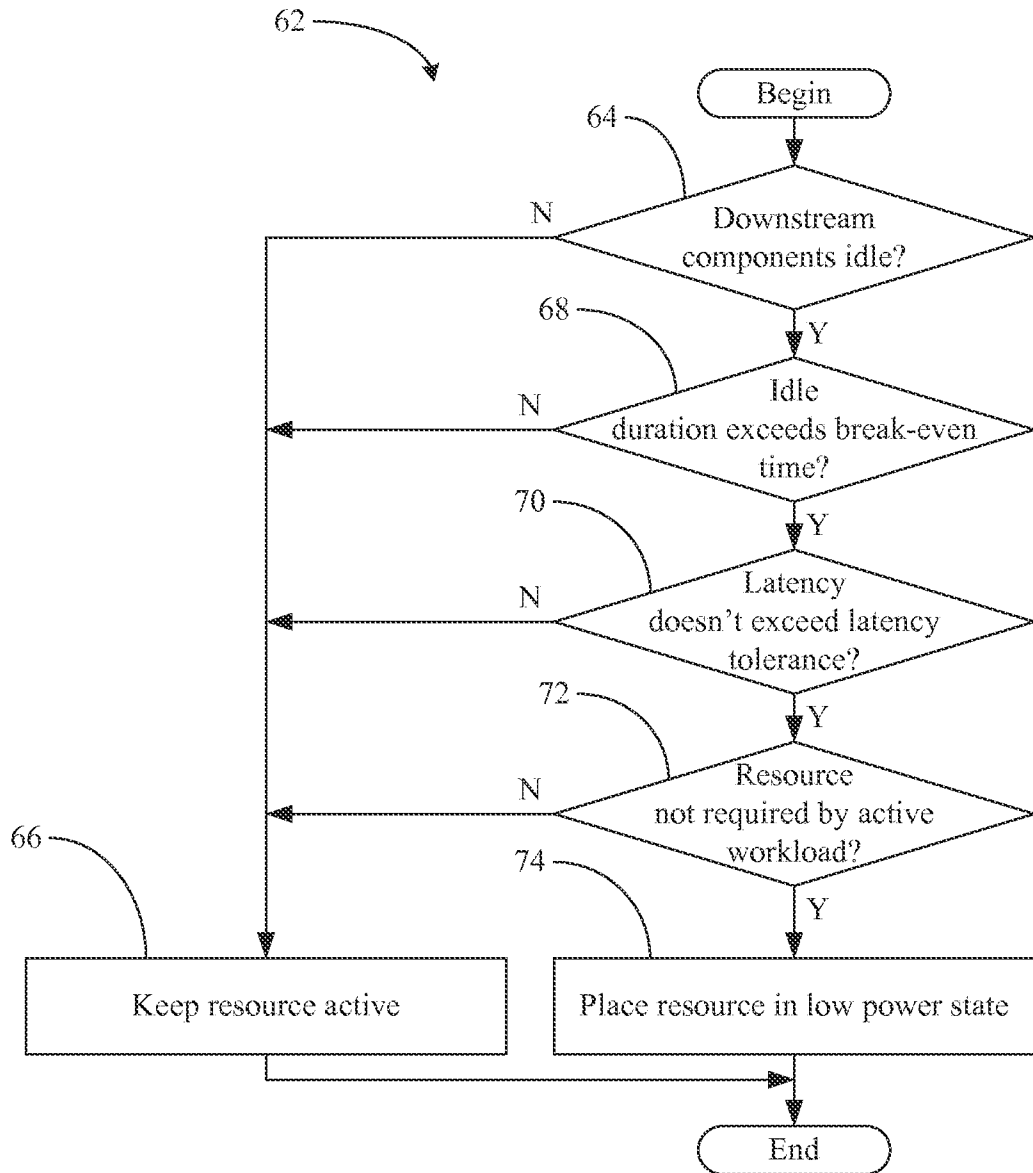


FIG. 6

## INTERNATIONAL SEARCH REPORT

International application No.  
**PCT/US2012/066656****A. CLASSIFICATION OF SUBJECT MATTER****G06F 1/32(2006.01)i**

According to International Patent Classification (IPC) or to both national classification and IPC

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)

G06F 1/32; G06F 19/00; G06F 1/20; G06F 1/26; G06F 1/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS(KIPO internal) &amp; keywords: resource, idle, predict time, partial, and similar terms.

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y A	wo 2011-085330 AI (QUALCOMM INC.) 14 July 2011 See paragraphs [0004] , [0027] , [0033] , [0054] , [0063] , and [0065] ; claims 1 and 3; and figure 4.	1-2 , 8-9 , 15-16 ,22-23 3-7 , 10-14 , 17-21 ,24-28
Y A	US 8024590 B2 (SONG, JUSTIN et al.) 20 September 2011 See col . 3 , lines 49-55 , col . 6 , lines 8-13 ; and claim 1.	1-2 , 8-9 , 15-16 ,22-23
A	US 2010-0115314 AI (SULTENFUSS, ANDREW T.) 6 May 2010 See paragraphs [0020]-[0021] ; and figure 1.	1-28
A	US 2010-0287396 AI (BARTH, ANDY et al.) 11 November 2010 See paragraphs [0025]-[0028] ; and figure 1.	1-28
A	US 2007-0094525 AI (UGUEN, LAURENT et al.) 26 April 2007 See paragraphs [0023]-[0026] , and [0051] -[0054] ; claim 1; and figures 2 and 5.	1-28



Further documents are listed in the continuation of Box C.



See patent family annex.

\* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&amp;" document member of the same patent family

Date of the actual completion of the international search

28 FEBRUARY 2013 (28.02.2013)

Date of mailing of the international search report

**28 FEBRUARY 2013 (28.02.2013)**

Name and mailing address of the ISA/KR



Facsimile No. 82-42-472-7140

Authorized officer

KIM, Byoung Sung

Telephone No. 82-42-481-8403



**INTERNATIONAL SEARCH REPORT**

Information on patent family members

International application No.

**PCT/US2012/066656**

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
Wo 2011-085330 A1	14.07.2011	KR 10-2012-0117015 A KR 10-2012-0117020 A US 2011-0173474 A1 US 2011-0173475 A1 Wo 2011-085333 A1	23. 10.,,2012 23. 10.,,2012 14. 07.,,2011 14. 07.,,2011 14. 07.,,2011
us 8024590 B2	20.09.2011	us 2009-0150695 A1 us 2012-0023355 A1	11. 06.,,2009 26. 01.,,2012
us 2010-0115314 A1	06.05.2010	us 2012-0036379 A1 us 8065540 B2	09. 02.,,2012 22. 11.,,2011
us 2010-0287396 A1	11. 11.2010	EP 2227727 A1 JP 2011-508328 A wo 2009-083753 A1	15. 09.,,2010 10. 03.,,2011 09. 07.,,2009
us 2007-0094525 A1	26.04.2007	EP 1677175 A2 EP 1677175 A3 US 7689839 B2	05. 07.,,2006 04. 05.,,2011 30. 03.,,2010