US 20070049133A1

(19) United States

(12) Patent Application Publication

(10) Pub. No.: US 2007/0049133 A1

(43) Pub. Date: Mar. 1, 2007

Conroy et al.

(54) METHODS AND APPARATUSES FOR DYNAMIC POWER CONTROL

(76) Inventors: **David G. Conroy**, El Granada, CA (US); **Michael Culbert**, Monte Sereno, CA (US); **Keith A. Cox**, Campbell, CA (US)

Correspondence Address:
BLAKELY SOKOLOFF TAYLOR & ZAFMAN
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1030 (US)

(21) Appl. No.: 11/327,275

(22) Filed: Jan. 5, 2006

Related U.S. Application Data

(63) Continuation-in-part of application No. 11/212,970, filed on Aug. 25, 2005.

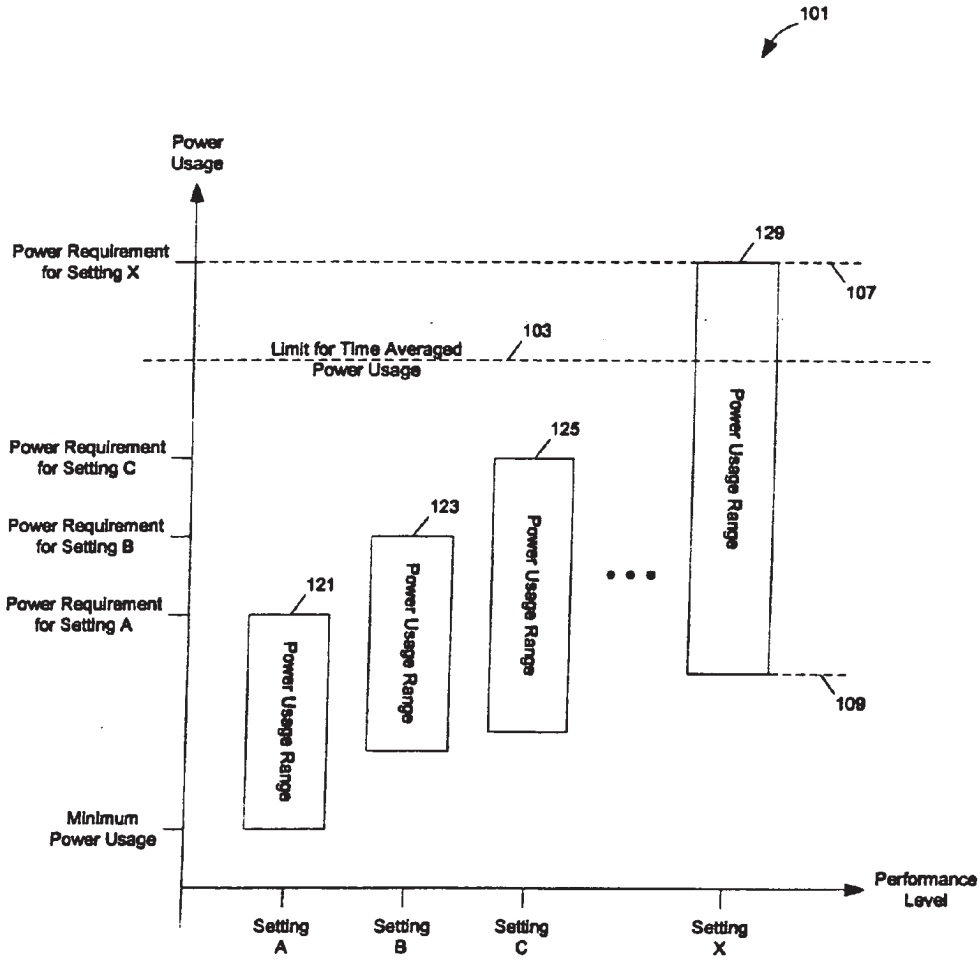
Publication Classification

(51) Int. Cl.
H01R 13/73 (2006.01)

(52) U.S. Cl. 439/894

(57) ABSTRACT

Exemplary embodiments of methods and apparatuses to dynamically redistribute power in a system that includes a plurality of subsystems are described. A load profile of the system is identified. The power is redistributed between the subsystems while tracking the load profile. The load profile may be an asymmetric, or a balanced load profile. The load profile is identified based on a utilization factor for each of the subsystems. In one embodiment, the power used by each of the subsystems is sensed by one or more sensors or predicted or estimated. A utilization factor, which may be a ratio of the actual power used by the subsystem to the power allocated to the subsystem, is calculated. The load profile is determined using the utilization factor of each of the subsystems. A power weighting arrangement between the subsystems, for example, a power distribution table, is selected based on the load profile.



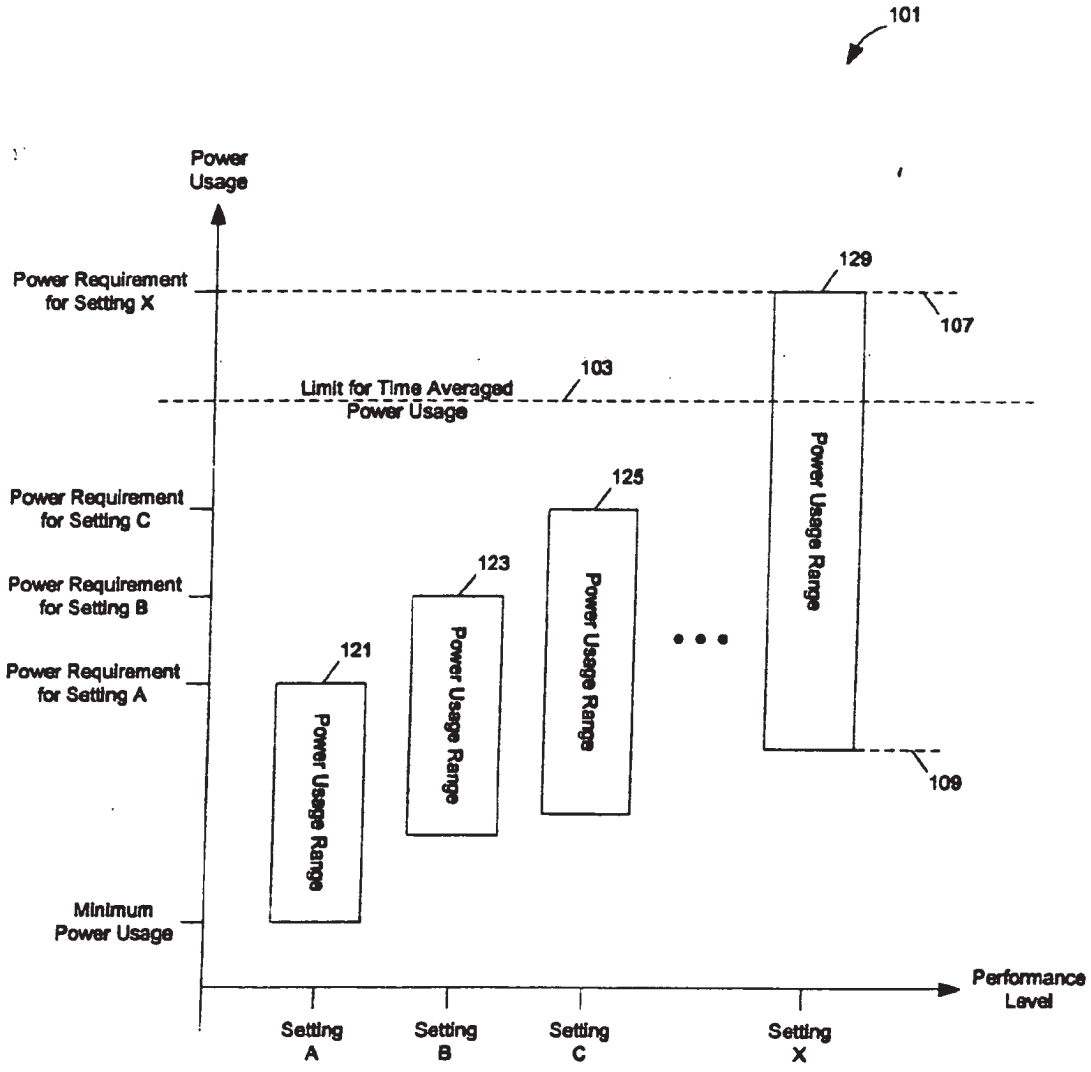

101
Power Usage
Power Requirement for Setting X
129
107
Limit for Time Averaged Power Usage
103
Power Requirement for Setting C
125
Power Usage Range
Power Requirement for Setting B
123
Power Usage Range
Power Requirement for Setting A
121
Power Usage Range
Power Usage Range
109
Minimum Power Usage
Performance Level
Setting A
Setting B
Setting C
Setting X

Fig. 1

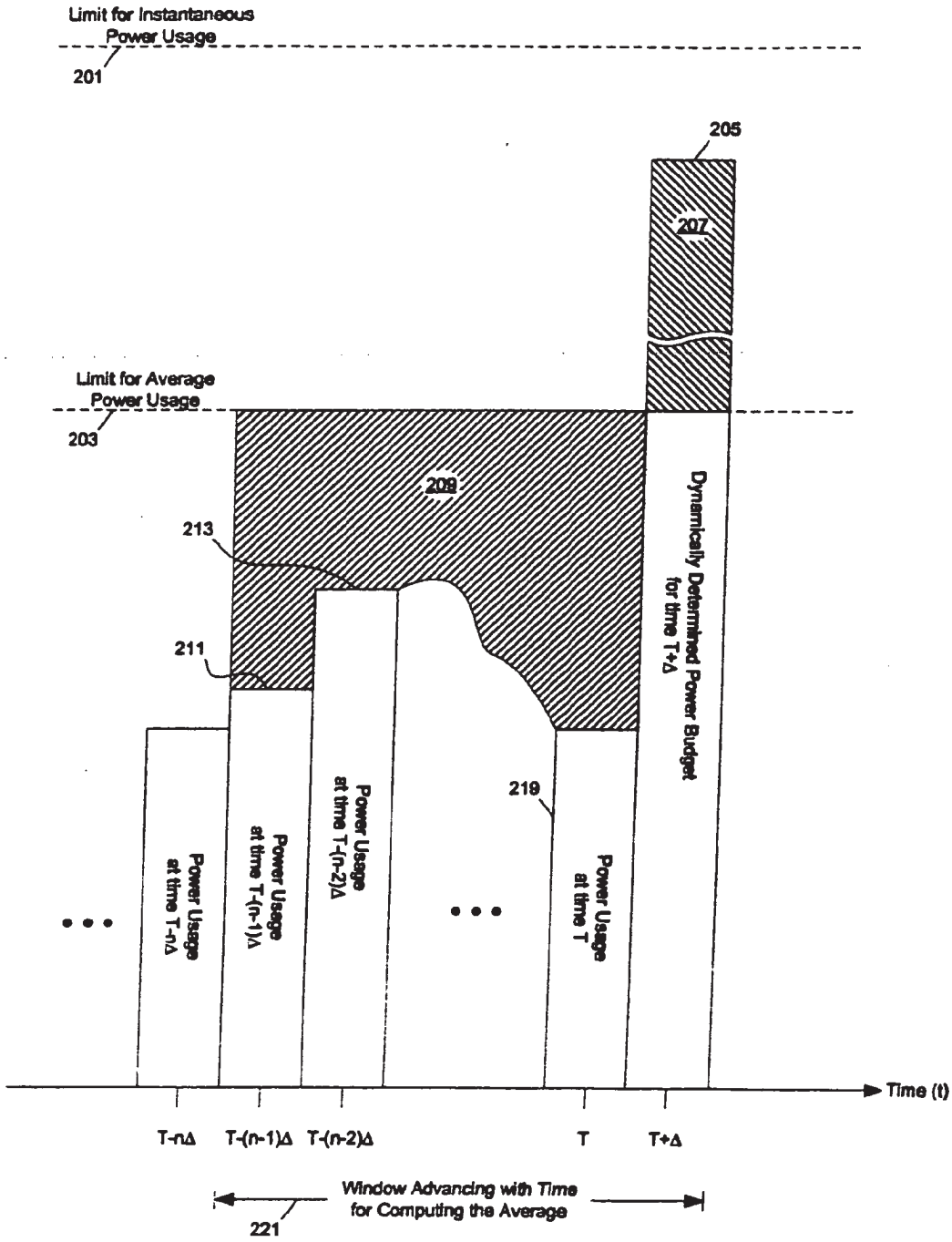

Limit for Instantaneous Power Usage
201
205
207
Limit for Average Power Usage
203
209
213
211
219
Dynamically Determined Power Budget for time T+Δ
Power Usage at time T-nΔ
Power Usage at time T-(n-1)Δ
Power Usage at time T-(n-2)Δ
Power Usage at time T
Time (t)
T-nΔ
T-(n-1)Δ
T-(n-2)Δ
T
T+Δ
Window Advancing with Time for Computing the Average
221

Fig. 2

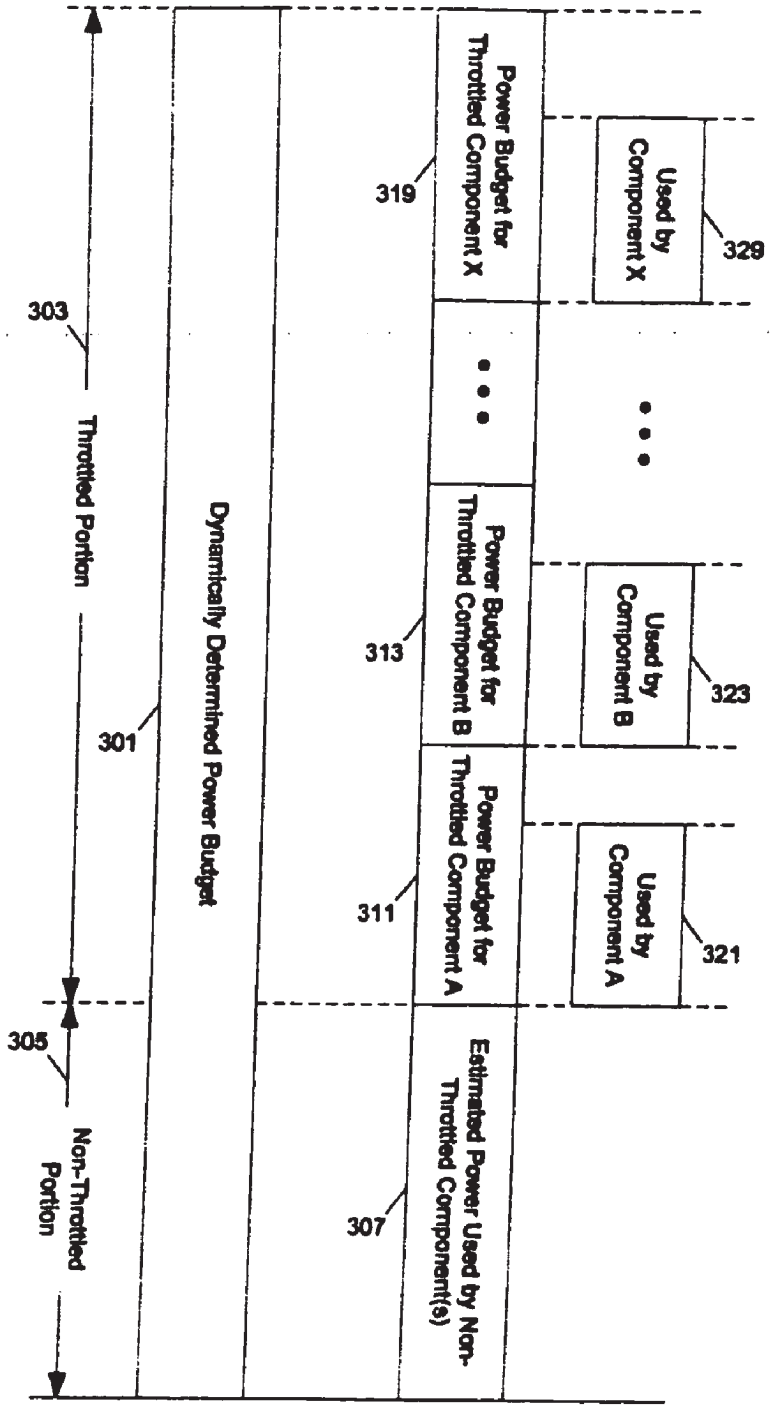

Used by Component X
329
Used by Component B
323
Used by Component A
321
Power Budget for Throttled Component X
319
Power Budget for Throttled Component B
313
Power Budget for Throttled Component A
311
Estimated Power Used by Non-Throttled Component(s)
307
Dynamically Determined Power Budget
301
303
Throttled Portion
305
Non-Throttled Portion

Fig. 3

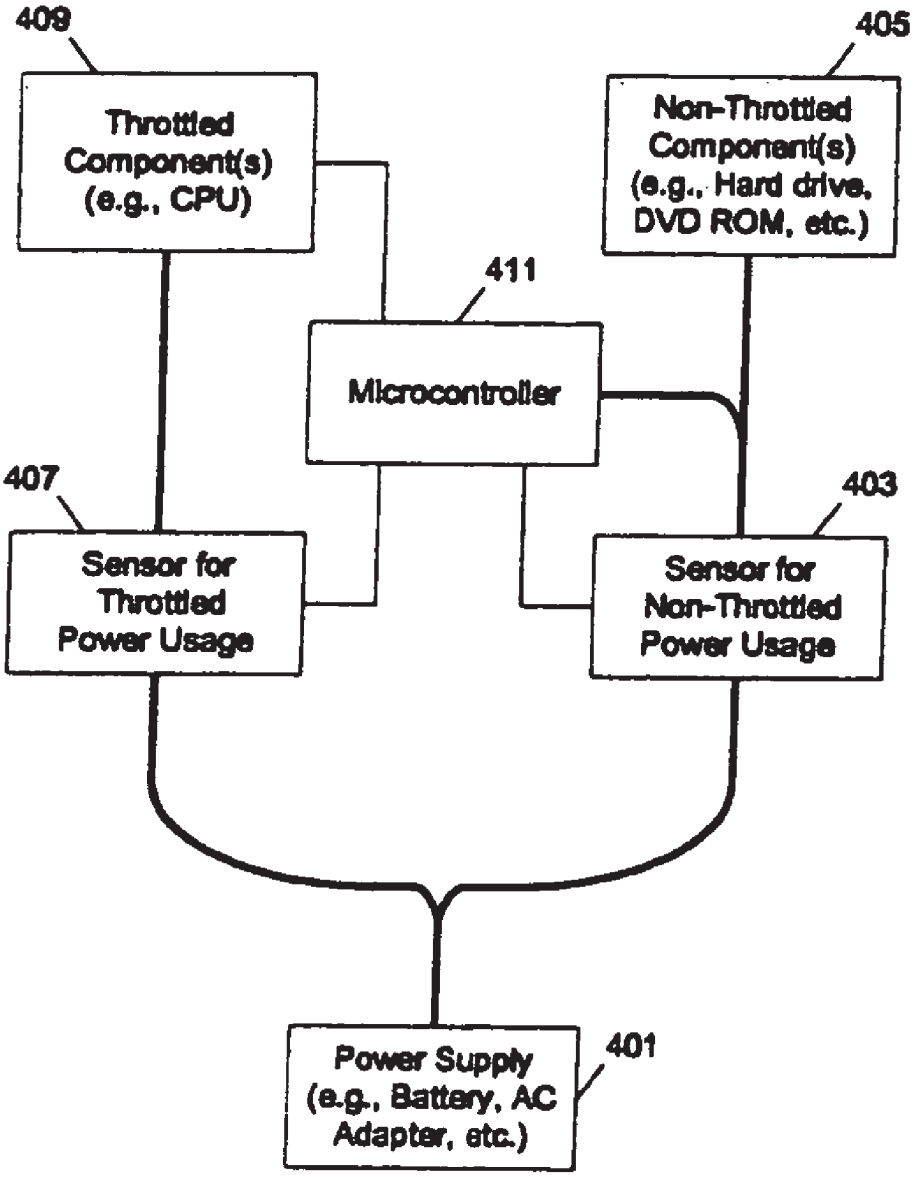

409
Throttled Component(s) (e.g., CPU)
405
Non-Throttled Component(s) (e.g., Hard drive, DVD ROM, etc.)
411
Microcontroller
407
Sensor for Throttled Power Usage
403
Sensor for Non-Throttled Power Usage
Power Supply (e.g., Battery, AC Adapter, etc.)
401

Fig. 4

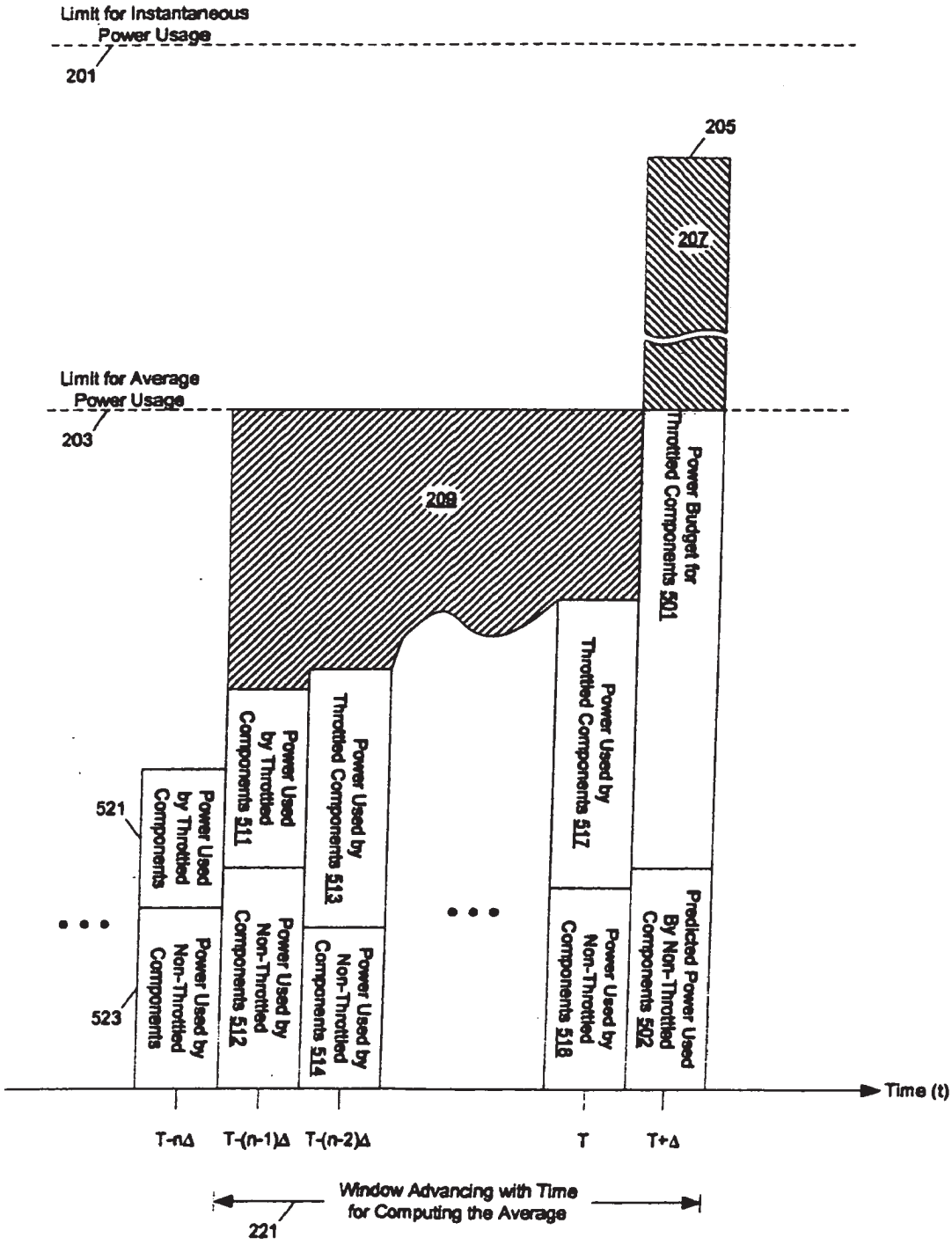

Limit for Instantaneous Power Usage
201
205
207
Limit for Average Power Usage
203
209
Power Budget for Throttled Components 501
Power Used by Throttled Components 511
Power Used by Throttled Components 513
Power Used by Throttled Components 517
521
Power Used by Throttled Components
Power Used by Non-Throttled Components 512
Power Used by Non-Throttled Components 514
Power Used by Non-Throttled Components 518
Predicted Power Used By Non-Throttled Components 502
523
Power Used by Non-Throttled Components
Time (t)
T-nΔ
T-(n-1)Δ
T-(n-2)Δ
T
T+Δ
Window Advancing with Time for Computing the Average
221

Fig. 5

601

Increasing Performance Level

Throttle Setting 1	CPU Core Voltage A	CPU Core Frequency A	Power Usage Requirement 1
Throttle Setting 2	CPU Core Voltage A	CPU Core Frequency B	Power Usage Requirement 2
Throttle Setting 3	CPU Core Voltage B	CPU Core Frequency B	Power Usage Requirement 3
⋮			
Throttle Setting n	CPU Core Voltage X	CPU Core Frequency Y	Power Usage Requirement N

Fig. 6

701

Increasing in a Target Goal Level

Global Throttle Setting 1	Setting 1 for Component A	• • •	Setting 1 for Component X	Power Usage Requirement 1
Global Throttle Setting 2	Setting 1 for Component A	• • •	Setting 1 for Component X	Power Usage Requirement 2
Global Throttle Setting 3	Setting 2 for Component A	• • •	Setting 1 for Component X	Power Usage Requirement 3
		⋮		
Global Throttle Setting n	Setting m for Component A	• • •	Setting q for Component X	Power Usage Requirement N

Fig. 7

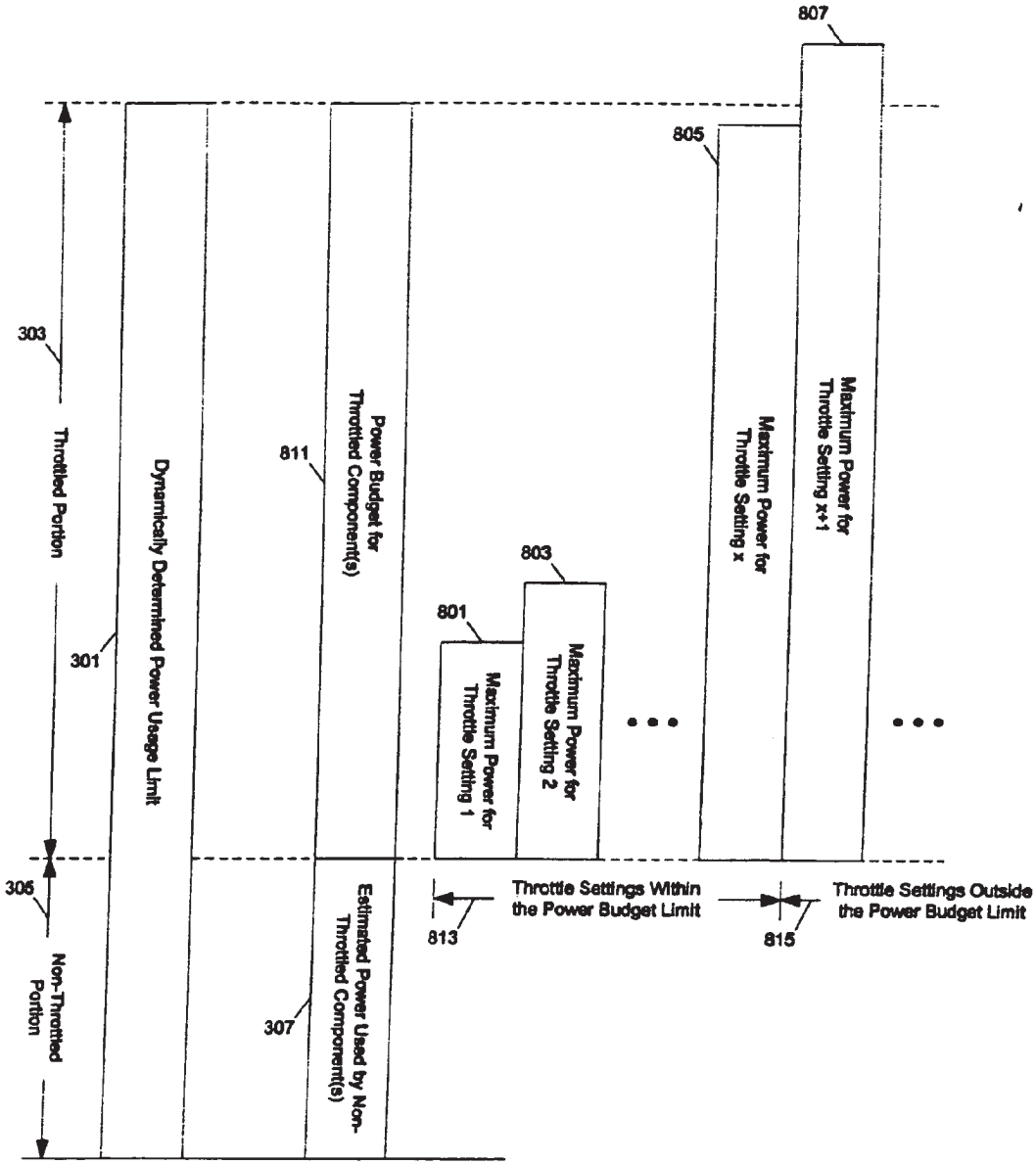

807
Maximum Power for Throttle Setting x+1
805
Maximum Power for Throttle Setting x
803
Maximum Power for Throttle Setting 2
801
Maximum Power for Throttle Setting 1
Throttle Settings Outside the Power Budget Limit
815
Throttle Settings Within the Power Budget Limit
813
Power Budget for Throttled Component(s)
811
Estimated Power Used by Non-Throttled Component(s)
307
Dynamically Determined Power Usage Limit
301
Throttled Portion
303
Non-Throttled Portion
305

Fig. 8

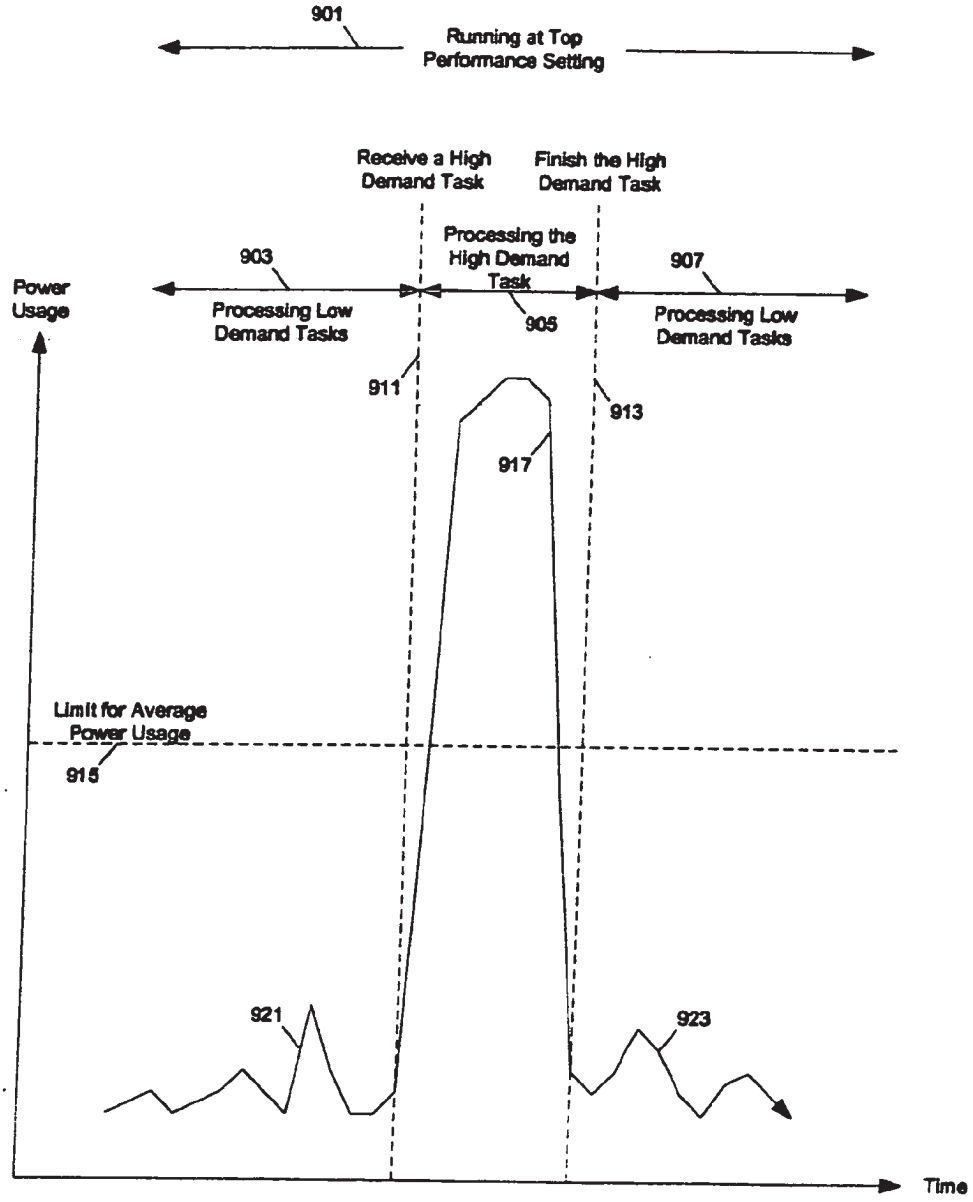

901
Running at Top Performance Setting
Receive a High Demand Task
Finish the High Demand Task
Processing the High Demand Task
903
907
Power Usage
Processing Low Demand Tasks
905
Processing Low Demand Tasks
911
913
917
Limit for Average Power Usage
915
921
923
Time

Fig. 9

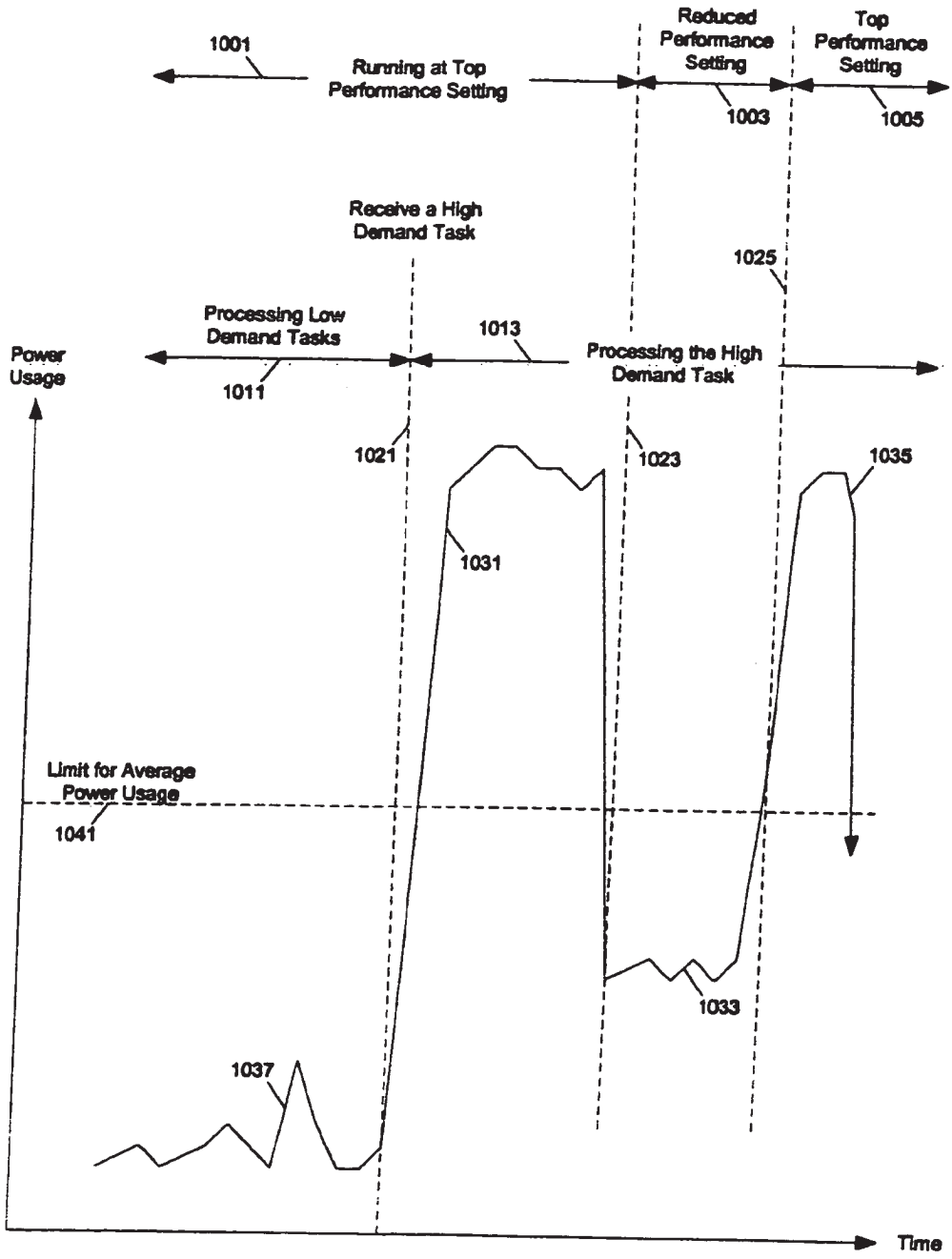

1001
Running at Top Performance Setting
Reduced Performance Setting
Top Performance Setting
1003
1005
Receive a High Demand Task
1025
Processing Low Demand Tasks
1013
Processing the High Demand Task
Power Usage
1011
1021
1023
1035
1031
Limit for Average Power Usage
1041
1033
1037
Time

Fig. 10

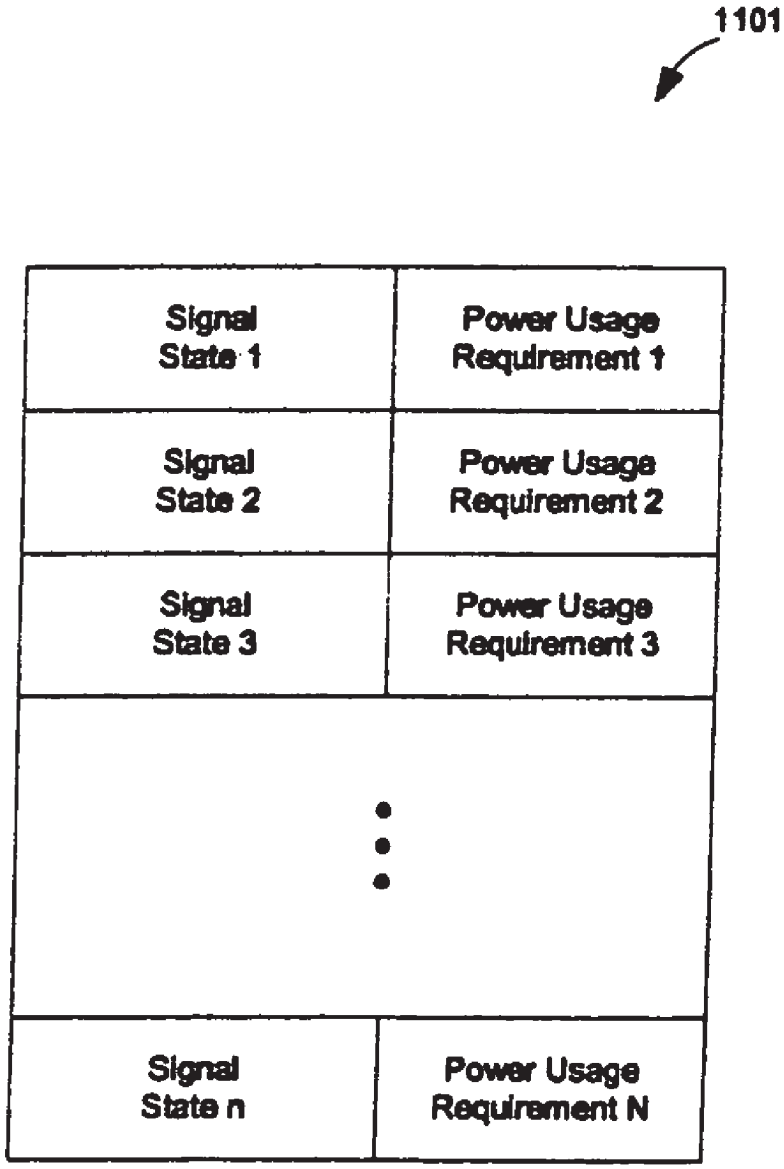

1101
Signal State 1
Power Usage Requirement 1
Signal State 2
Power Usage Requirement 2
Signal State 3
Power Usage Requirement 3
Signal State n
Power Usage Requirement N

Fig. 11

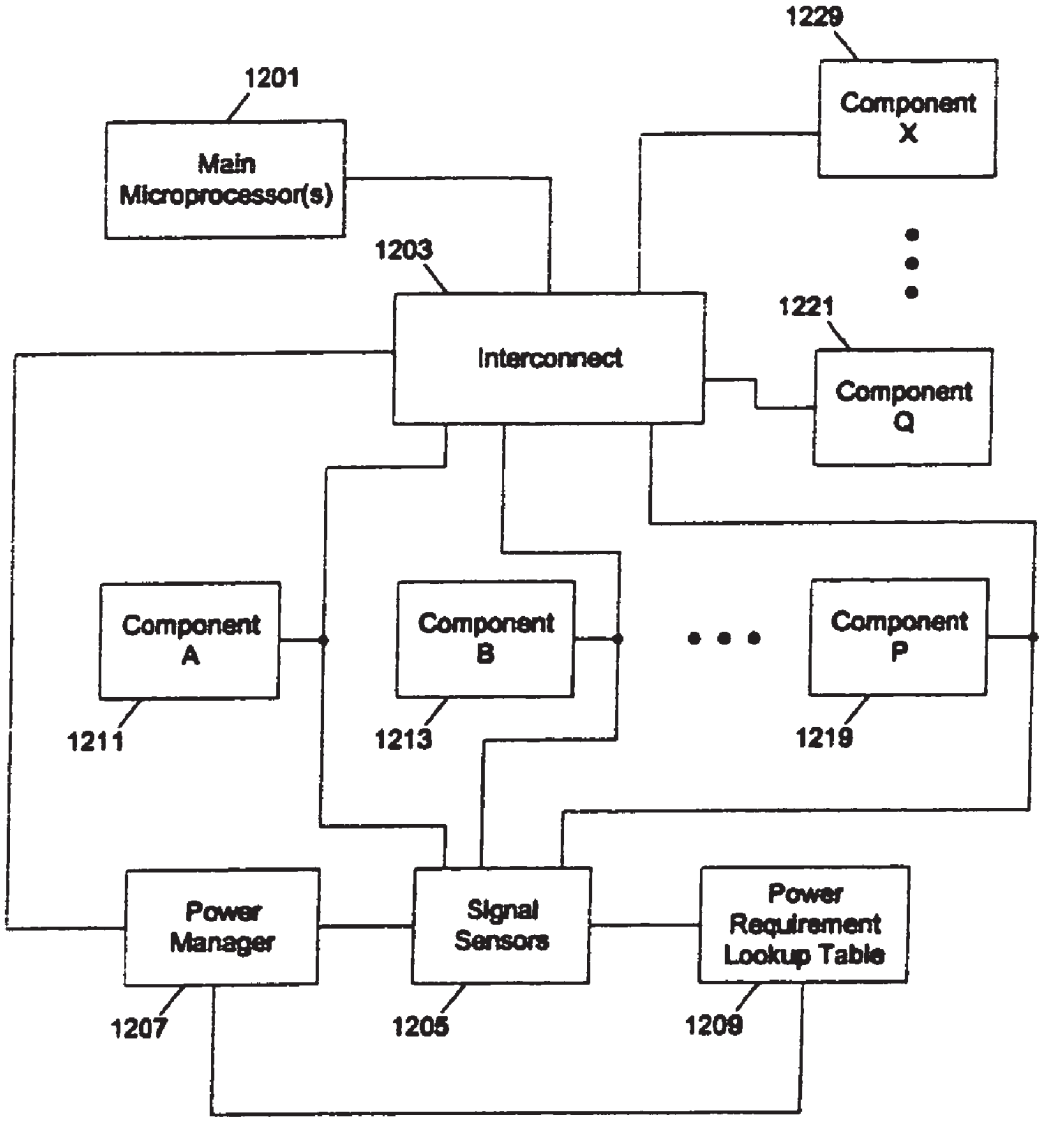

1229
Component X
1201
Main Microprocessor(s)
1203
Interconnect
1221
Component Q
Component A
Component B
Component P
1211
1213
1219
Power Manager
Signal Sensors
Power Requirement Lookup Table
1207
1205
1209

Fig. 12

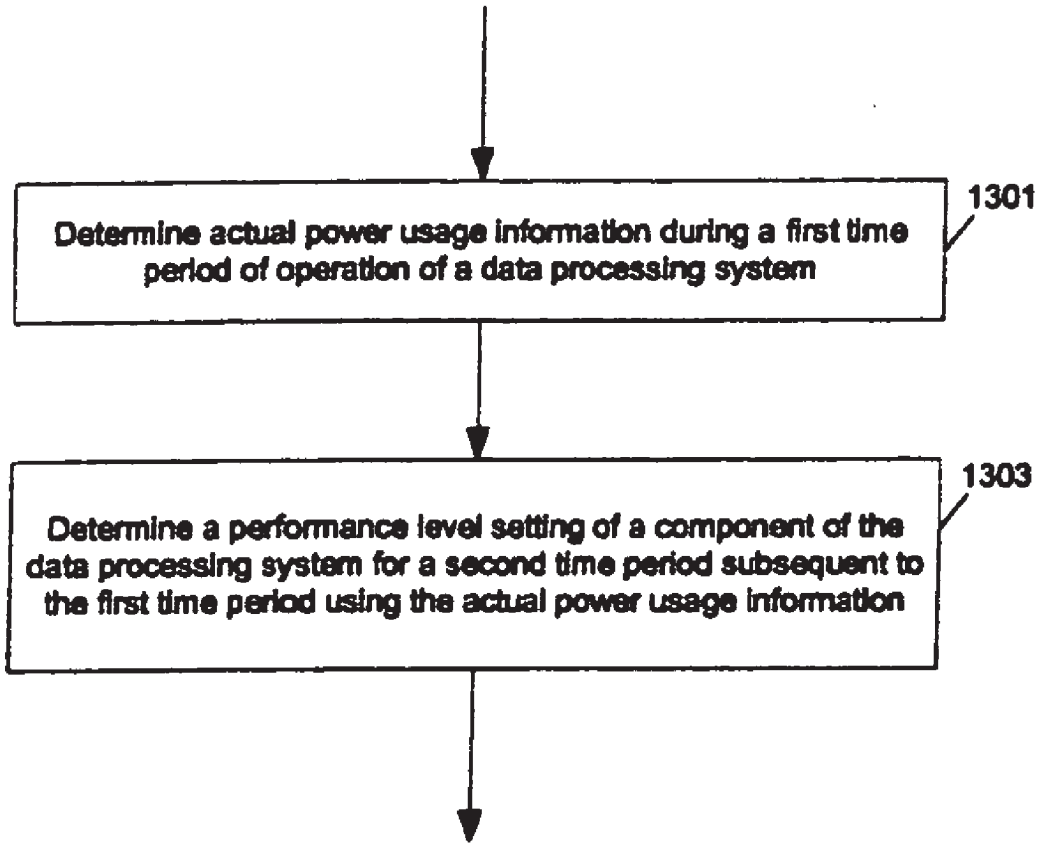

Determine actual power usage information during a first time period of operation of a data processing system
1301
Determine a performance level setting of a component of the data processing system for a second time period subsequent to the first time period using the actual power usage information
1303

Fig. 13

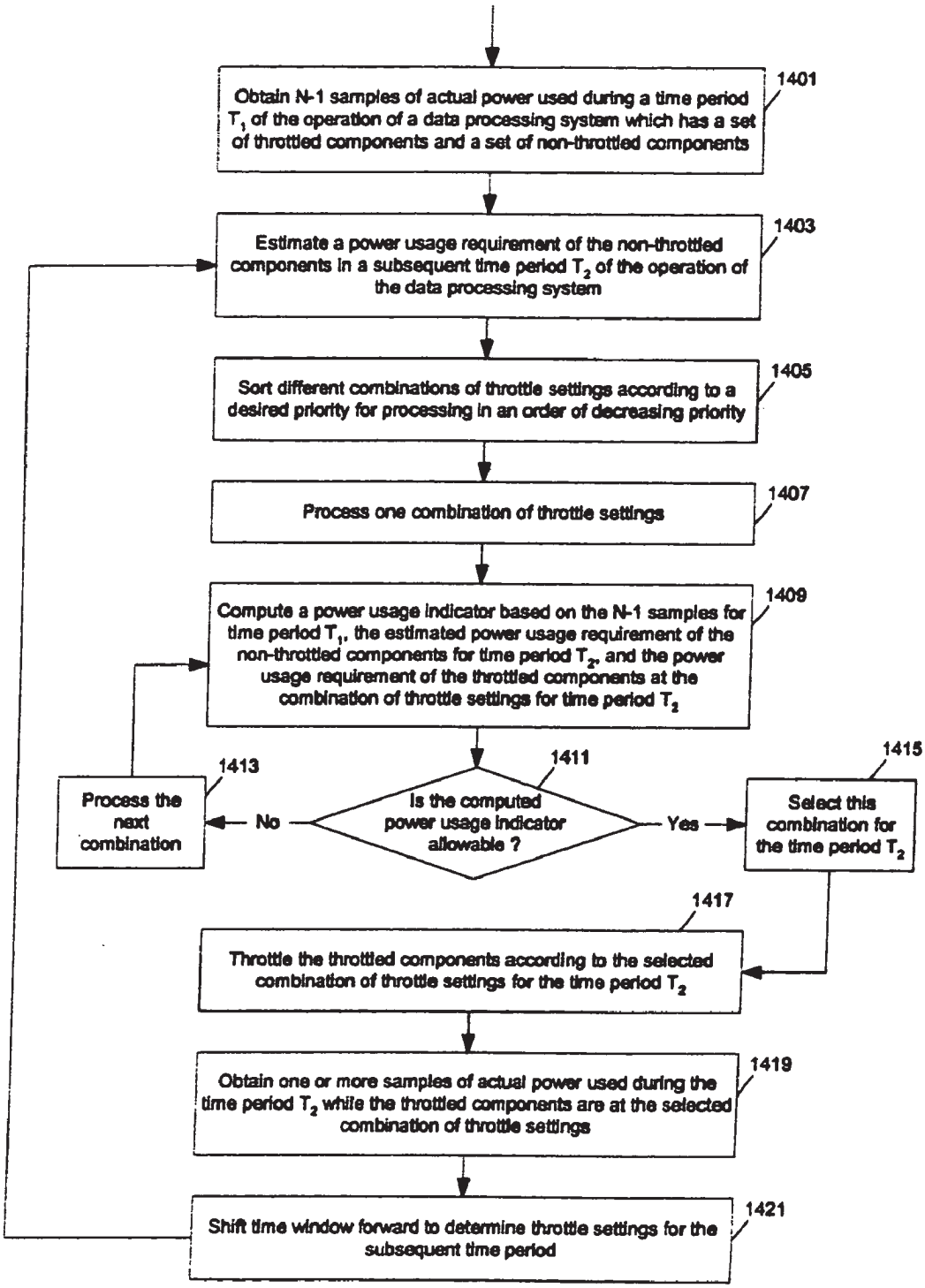

Obtain N-1 samples of actual power used during a time period T_1 of the operation of a data processing system which has a set of throttled components and a set of non-throttled components
1401
Estimate a power usage requirement of the non-throttled components in a subsequent time period T_2 of the operation of the data processing system
1403
Sort different combinations of throttle settings according to a desired priority for processing in an order of decreasing priority
1405
Process one combination of throttle settings
1407
Compute a power usage indicator based on the N-1 samples for time period T_1, the estimated power usage requirement of the non-throttled components for time period T_2, and the power usage requirement of the throttled components at the combination of throttle settings for time period T_2
1409
Is the computed power usage indicator allowable ?
1411
No
Yes
Process the next combination
1413
Select this combination for the time period T_2
1415
Throttle the throttled components according to the selected combination of throttle settings for the time period T_2
1417
Obtain one or more samples of actual power used during the time period T_2 while the throttled components are at the selected combination of throttle settings
1419
Shift time window forward to determine throttle settings for the subsequent time period
1421

Fig. 14

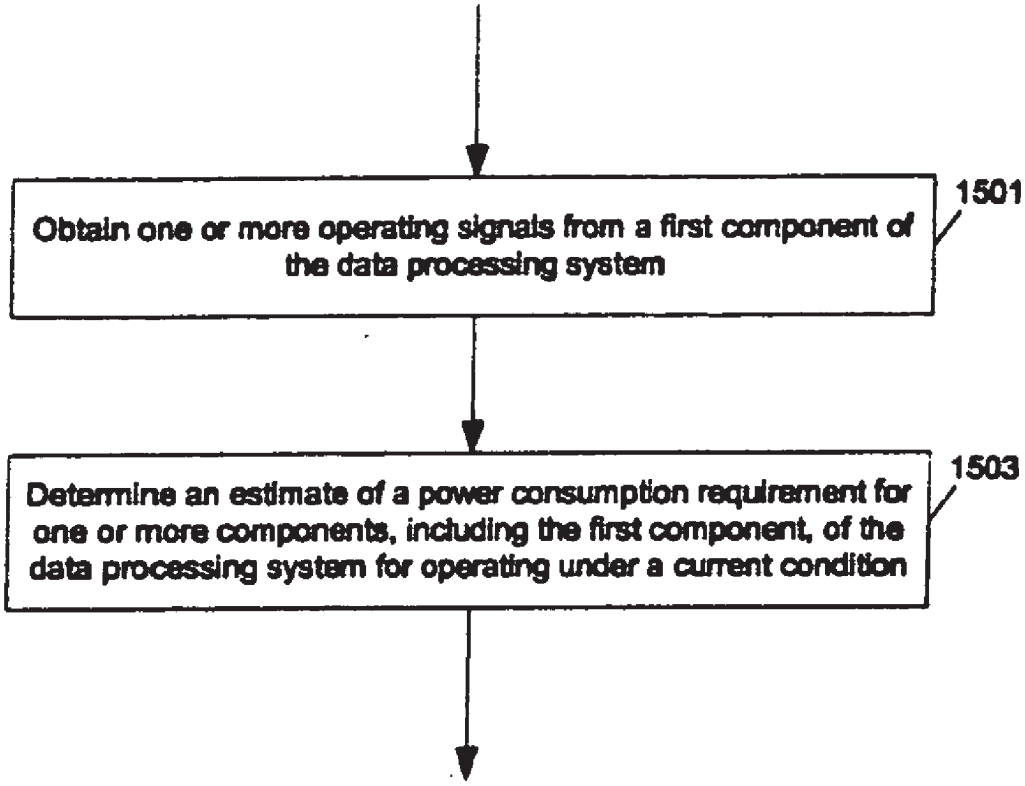

Obtain one or more operating signals from a first component of the data processing system
1501
Determine an estimate of a power consumption requirement for one or more components, including the first component, of the data processing system for operating under a current condition
1503

Fig. 15

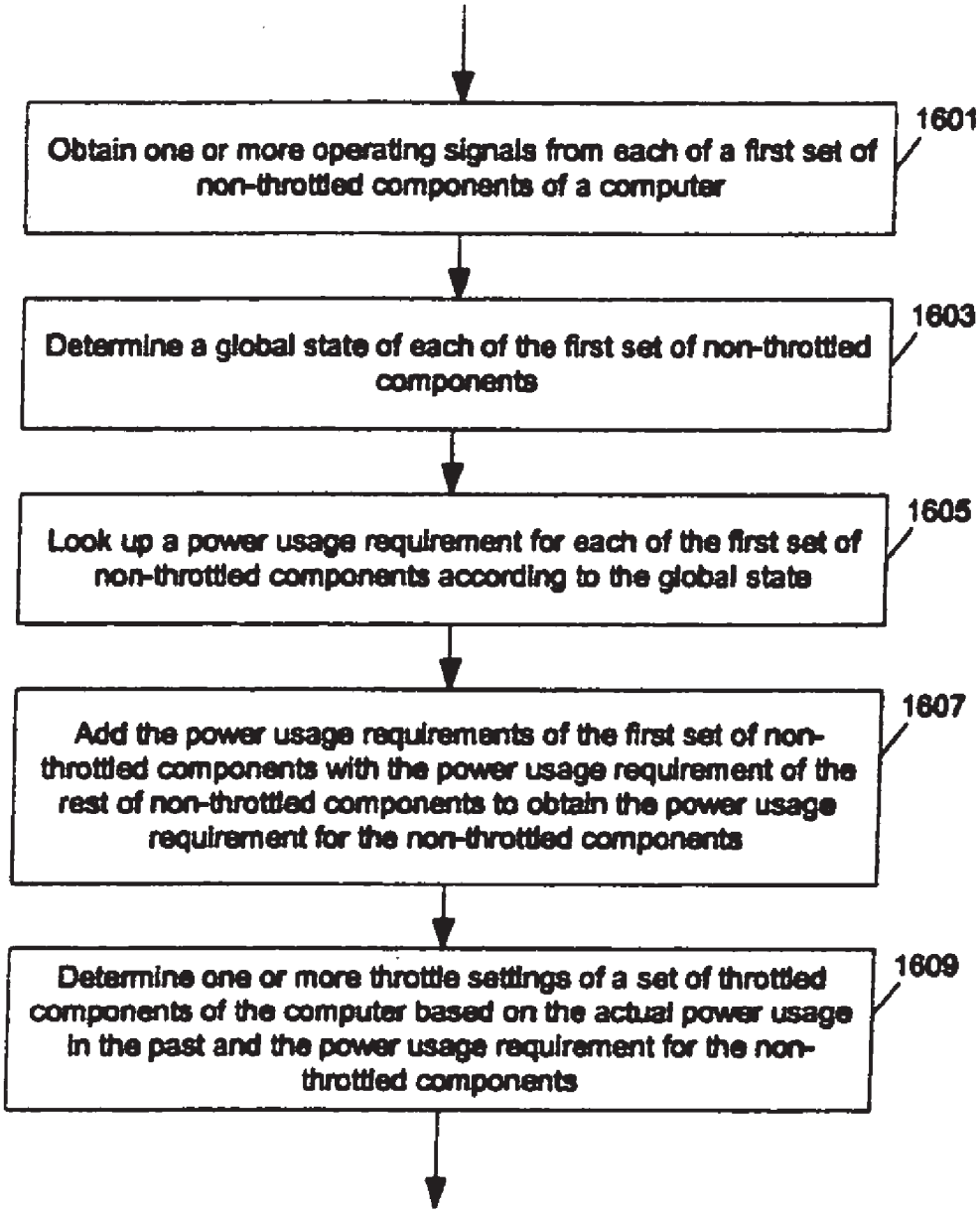

Obtain one or more operating signals from each of a first set of non-throttled components of a computer
1601
Determine a global state of each of the first set of non-throttled components
1603
Look up a power usage requirement for each of the first set of non-throttled components according to the global state
1605
Add the power usage requirements of the first set of non-throttled components with the power usage requirement of the rest of non-throttled components to obtain the power usage requirement for the non-throttled components
1607
Determine one or more throttle settings of a set of throttled components of the computer based on the actual power usage in the past and the power usage requirement for the non-throttled components
1609

Fig. 16

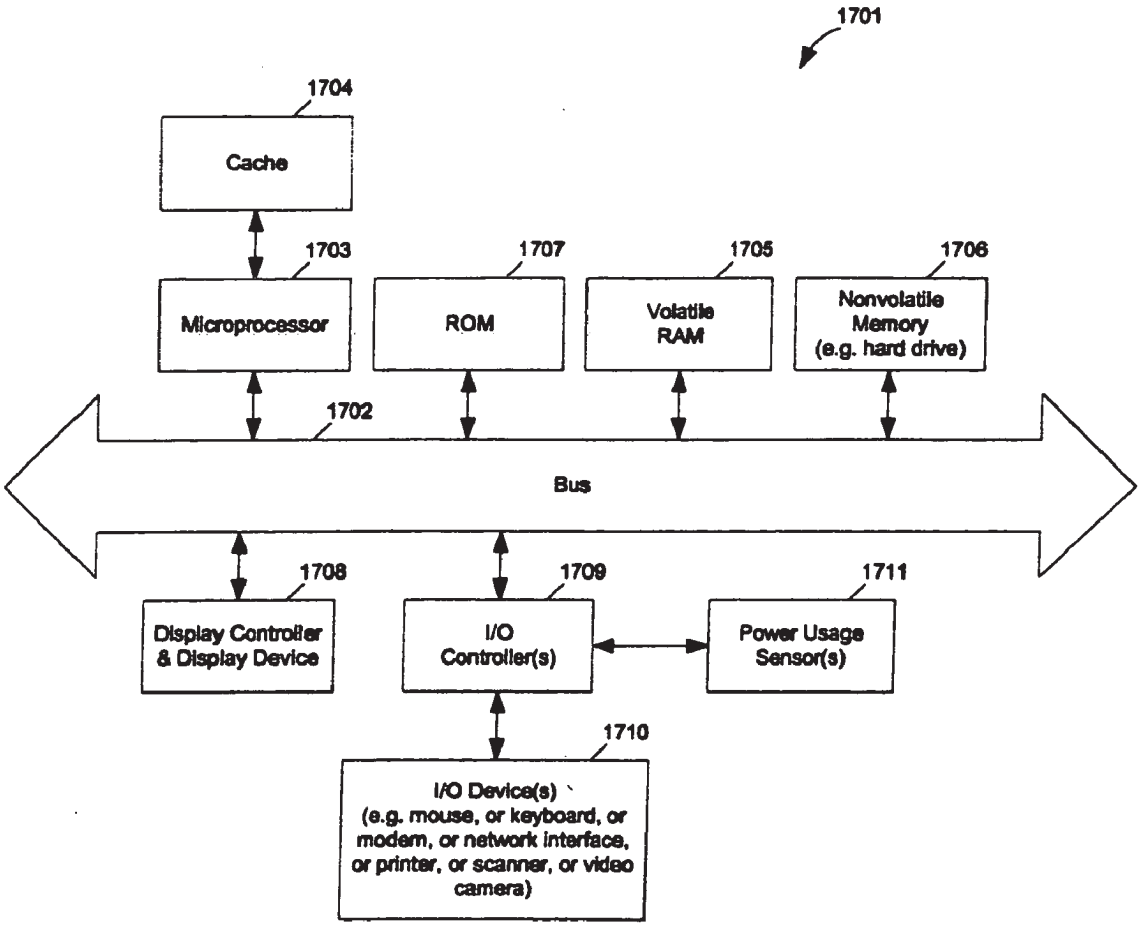
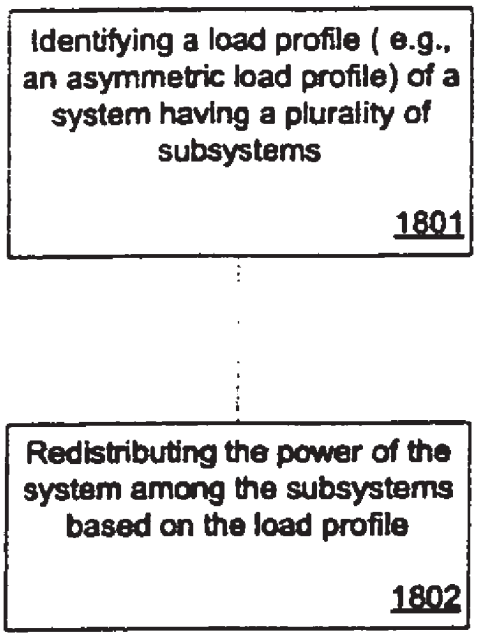

1701
1704
Cache
1703
Microprocessor
1707
ROM
1705
Volatile RAM
1706
Nonvolatile Memory (e.g. hard drive)
1702
Bus
1708
Display Controller & Display Device
1709
I/O Controller(s)
1711
Power Usage Sensor(s)
1710
I/O Device(s) (e.g. mouse, or keyboard, or modem, or network interface, or printer, or scanner, or video camera)

Fig. 17


Identifying a load profile (e.g., an asymmetric load profile) of a system having a plurality of subsystems
1801
Redistributing the power of the system among the subsystems based on the load profile
1802

1800

Fig. 18

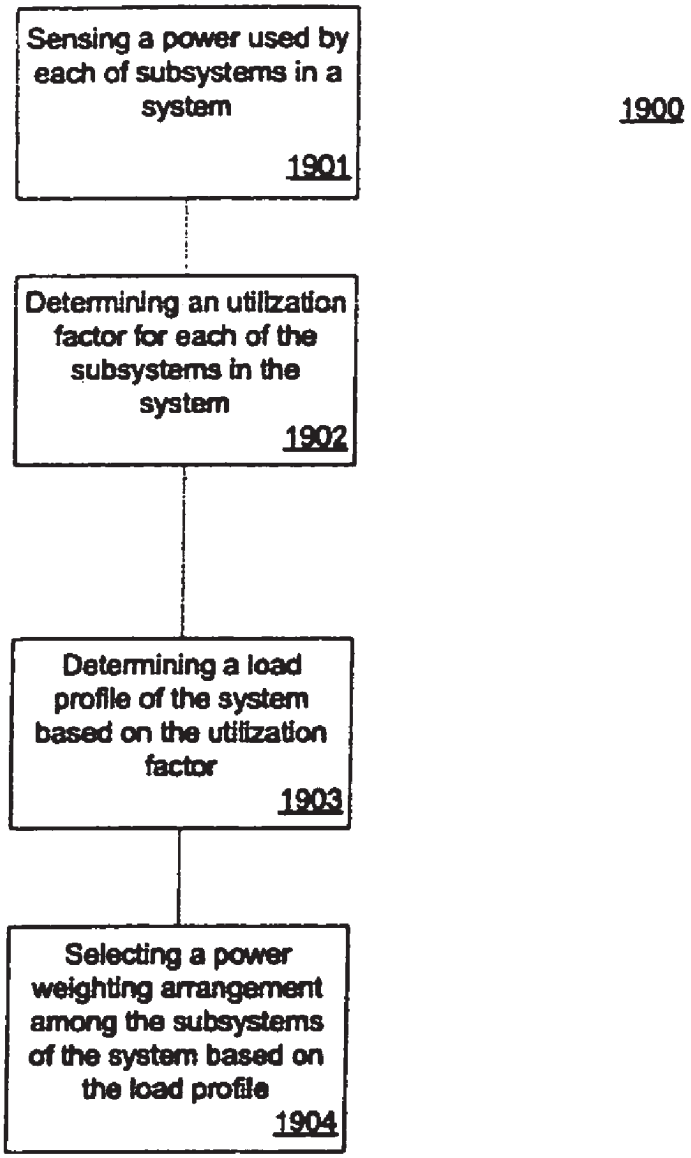

Sensing a power used by each of subsystems in a system
1901
1900
Determining an utilization factor for each of the subsystems in the system
1902
Determining a load profile of the system based on the utilization factor
1903
Selecting a power weighting arrangement among the subsystems of the system based on the load profile
1904

Fig. 19

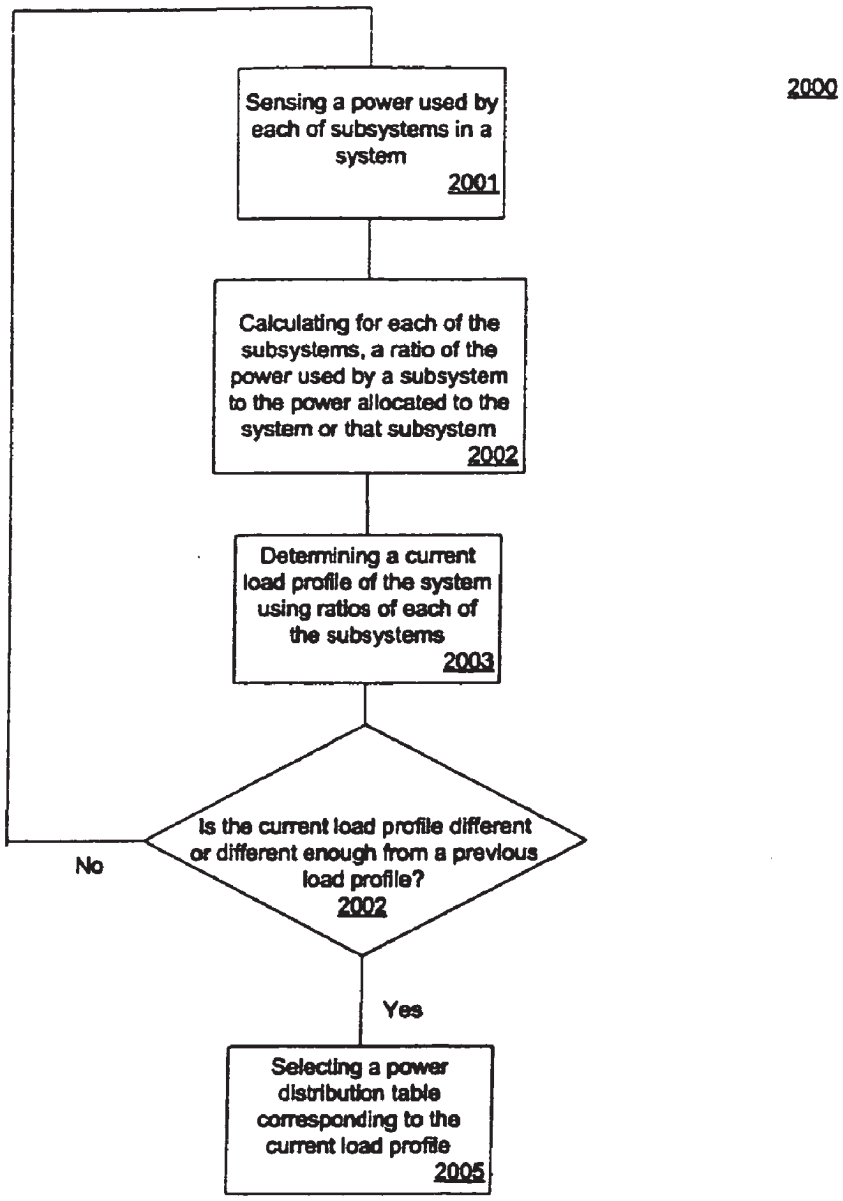

2000
Sensing a power used by each of subsystems in a system
2001
Calculating for each of the subsystems, a ratio of the power used by a subsystem to the power allocated to the system or that subsystem
2002
Determining a current load profile of the system using ratios of each of the subsystems
2003
Is the current load profile different or different enough from a previous load profile?
2002
No
Yes
Selecting a power distribution table corresponding to the current load profile
2005

Fig. 20

Balanced load profile K1 (e.g., 0.5, 0.5)

2100

2101 Settings	2102 CPU	2103 GPU	2104 Allocated Power
Setting 1	A1 (e.g., 10)	B1 (e.g., 10)	P1 (e.g., 20)
Setting 2	A2 (e.g., 7)	B2 (e.g., 7)	P2 (e.g., 14)
Setting 3	A3 (e.g., 5)	B3 (e.g., 5)	P3 (e.g., 10)

2105

Fig. 21 A

CPU-heavy load profile K2 (e.g., 0.75, 0.25)

2110

2101 Settings	2102 CPU	2103 GPU	2104 Allocated Power
Setting 1	A1 (e.g., 15)	B1 (e.g., 5)	P1 (e.g., 20)
Setting 2	A2 (e.g., 10)	B2 (e.g., 4)	P2 (e.g., 14)
Setting 3	A3 (e.g., 7)	B3 (e.g., 3)	P3 (e.g., 10)

2105

Fig. 21 B

GPU-heavy load profile K3 (e.g., 0.25, 0.75)

2120

2101 Settings	2102 CPU	2103 GPU	2104 Allocated Power
Setting 1	A1 (e.g., 5)	B1 (e.g., 15)	P1 (e.g., 20)
Setting 2	A2 (e.g., 3.5)	B2 (e.g., 10)	P2 (e.g., 14)
Setting 3	A3 (e.g., 3)	B3 (e.g., 7)	P3 (e.g., 10)

2105

Fig. 21 C

Load Profile Kn

2200

2206

2201 Settings	2202 Subsystem 1	2203 Subsystem 2	2204 Subsystem...	2205 Subsystem N	2206 Power
1	A1	B1	...	C1	P1
2	A2	B2	...	C2	P2
M	Am	Bm	...	Cm	Pm

Fig. 22

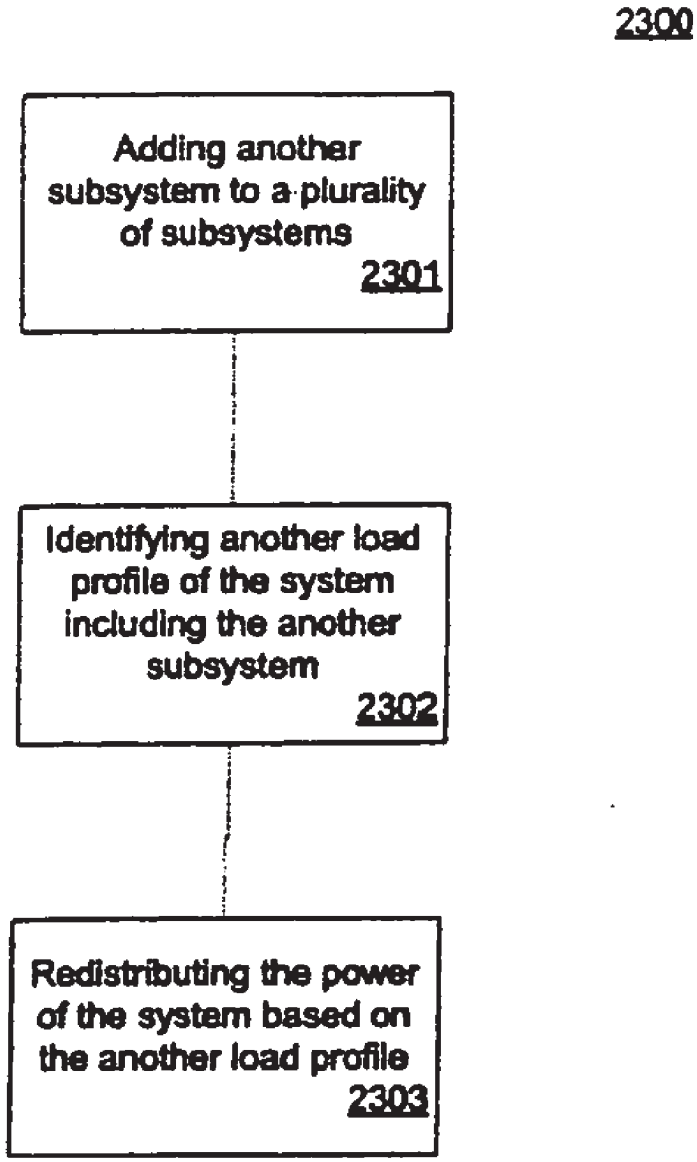

2300
Adding another subsystem to a plurality of subsystems
2301
Identifying another load profile of the system including the another subsystem
2302
Redistributing the power of the system based on the another load profile
2303

Fig. 23

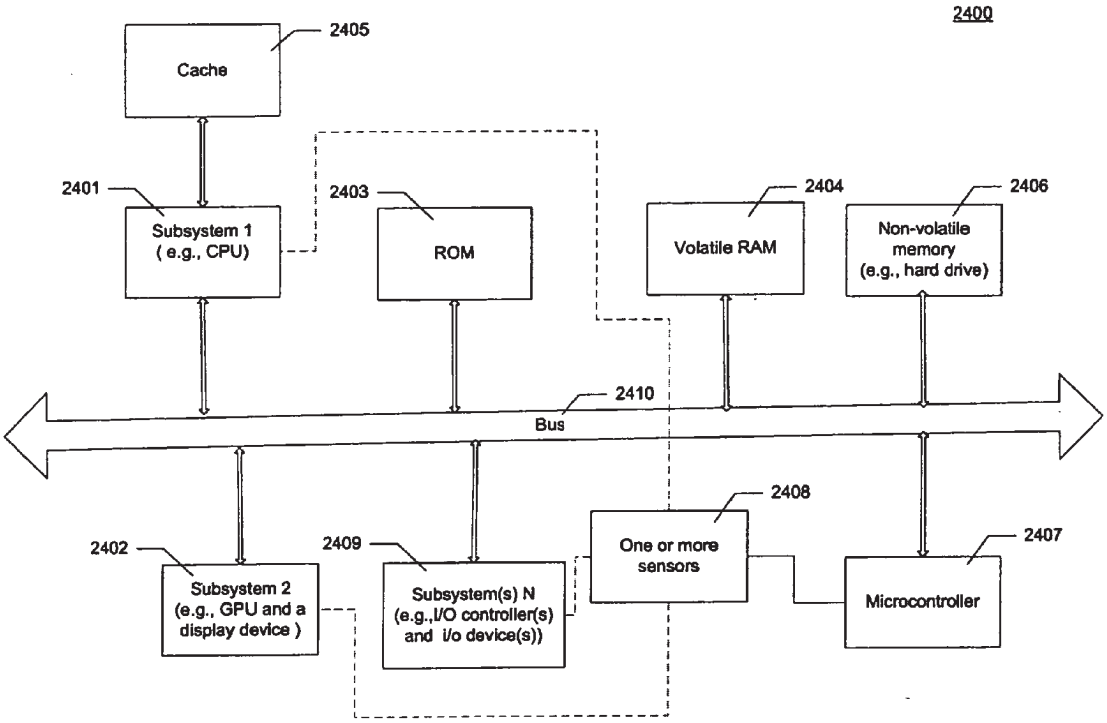

2400
2405
Cache
2401
Subsystem 1
(e.g., CPU)
2403
ROM
2404
Volatile RAM
2406
Non-volatile
memory
(e.g., hard drive)
2410
Bus
2408
One or more
sensors
2402
Subsystem 2
(e.g., GPU and a
display device)
2409
Subsystem(s) N
(e.g.,I/O controller(s)
and i/o device(s))
2407
Microcontroller

Fig. 24

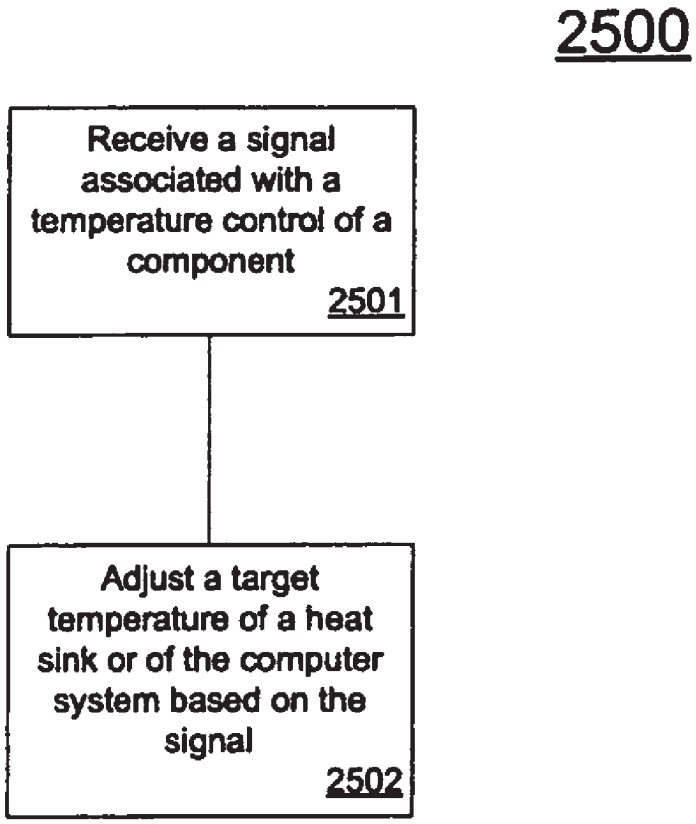

2500
Receive a signal associated with a temperature control of a component
2501
Adjust a target temperature of a heat sink or of the computer system based on the signal
2502

Fig. 25

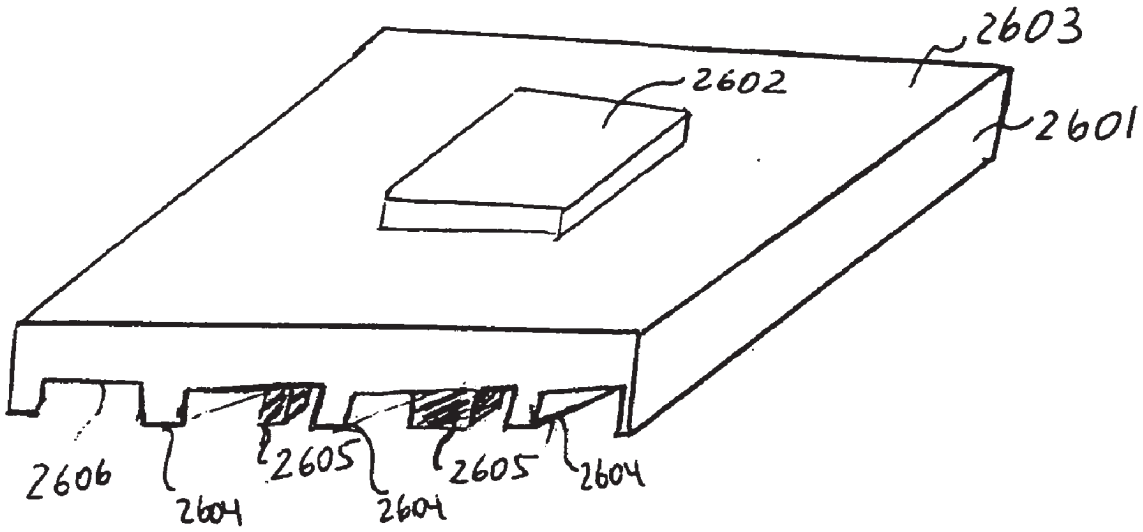


Fig. 26A

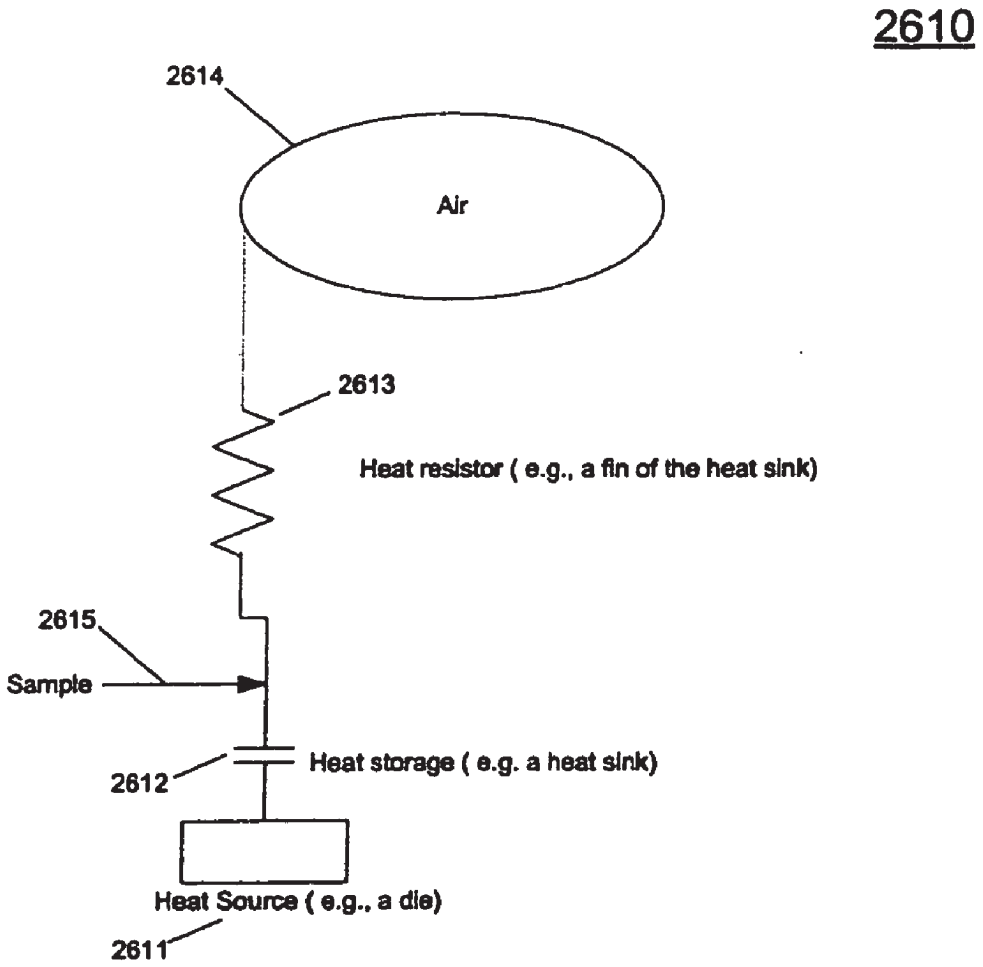

2610
2614
Air
2613
Heat resistor (e.g., a fin of the heat sink)
2615
Sample
2612
Heat storage (e.g. a heat sink)
Heat Source (e.g., a die)
2611

Fig. 26B

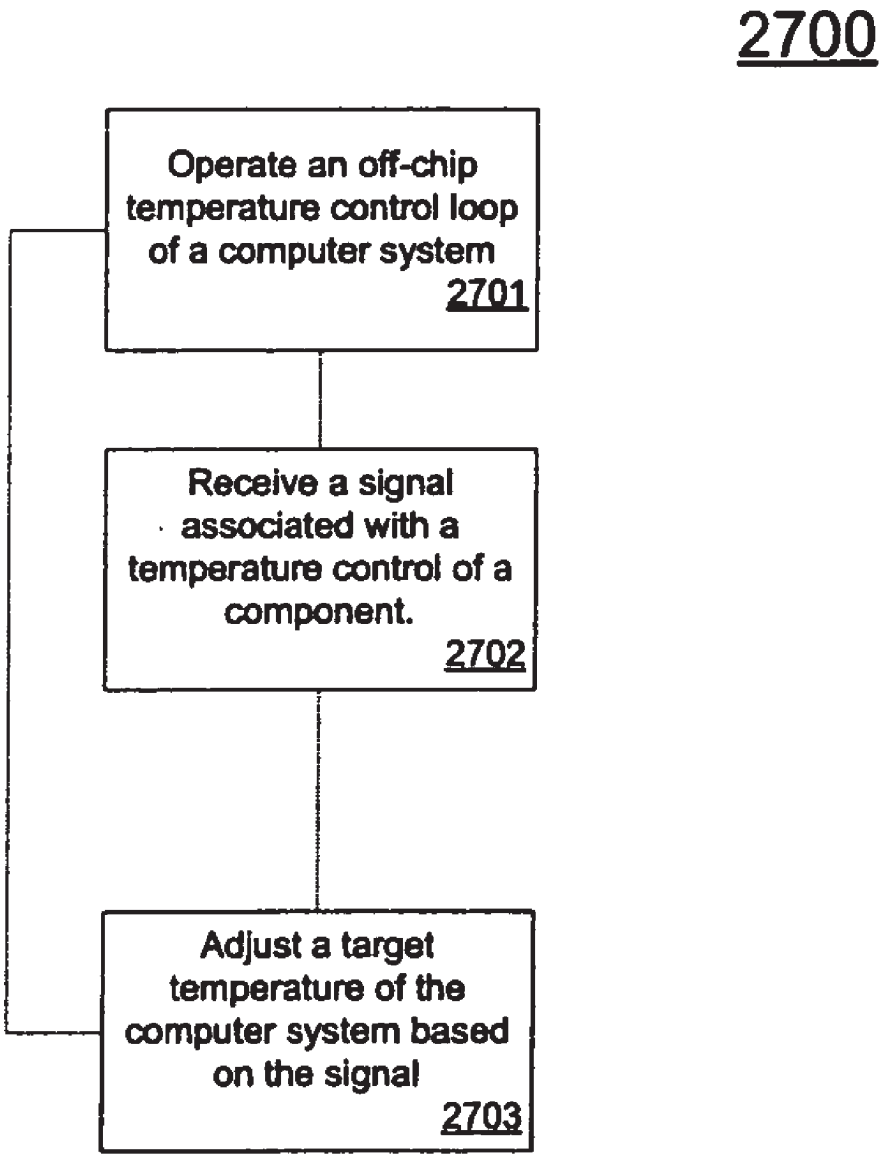

2700
Operate an off-chip temperature control loop of a computer system
2701
Receive a signal associated with a temperature control of a component.
2702
Adjust a target temperature of the computer system based on the signal
2703

Fig. 27

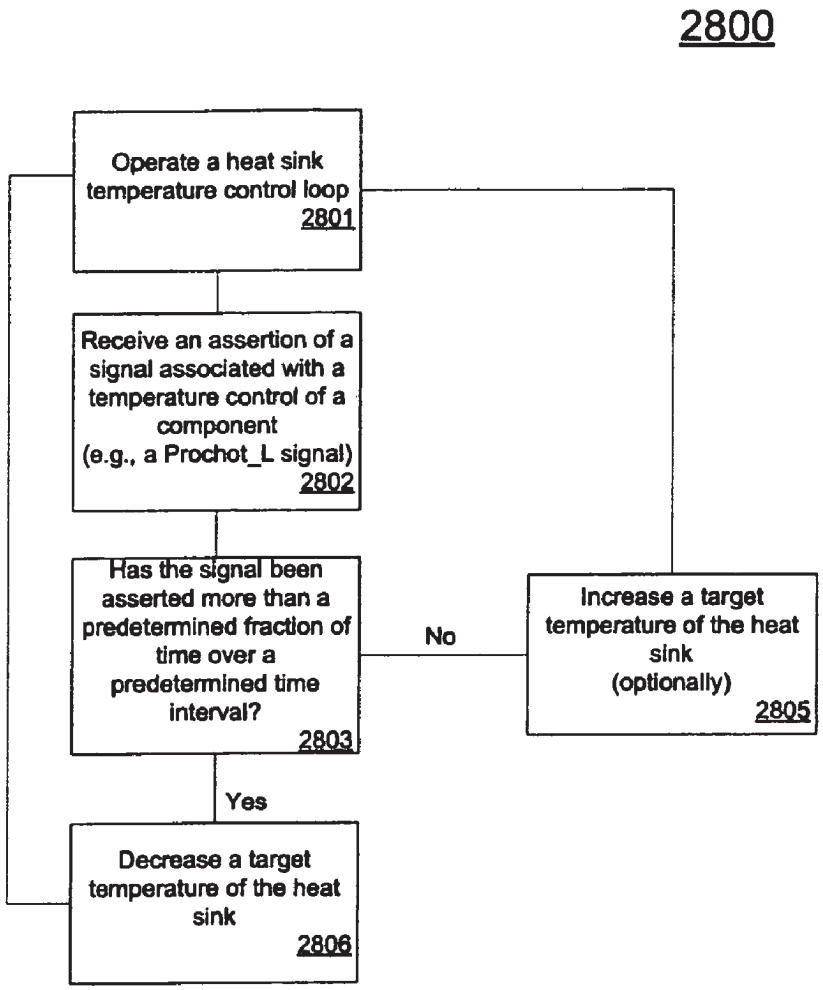

2800
Operate a heat sink temperature control loop
2801
Receive an assertion of a signal associated with a temperature control of a component (e.g., a Prochot_L signal)
2802
Has the signal been asserted more than a predetermined fraction of time over a predetermined time interval?
2803
No
Increase a target temperature of the heat sink (optionally)
2805
Yes
Decrease a target temperature of the heat sink
2806

Fig. 28

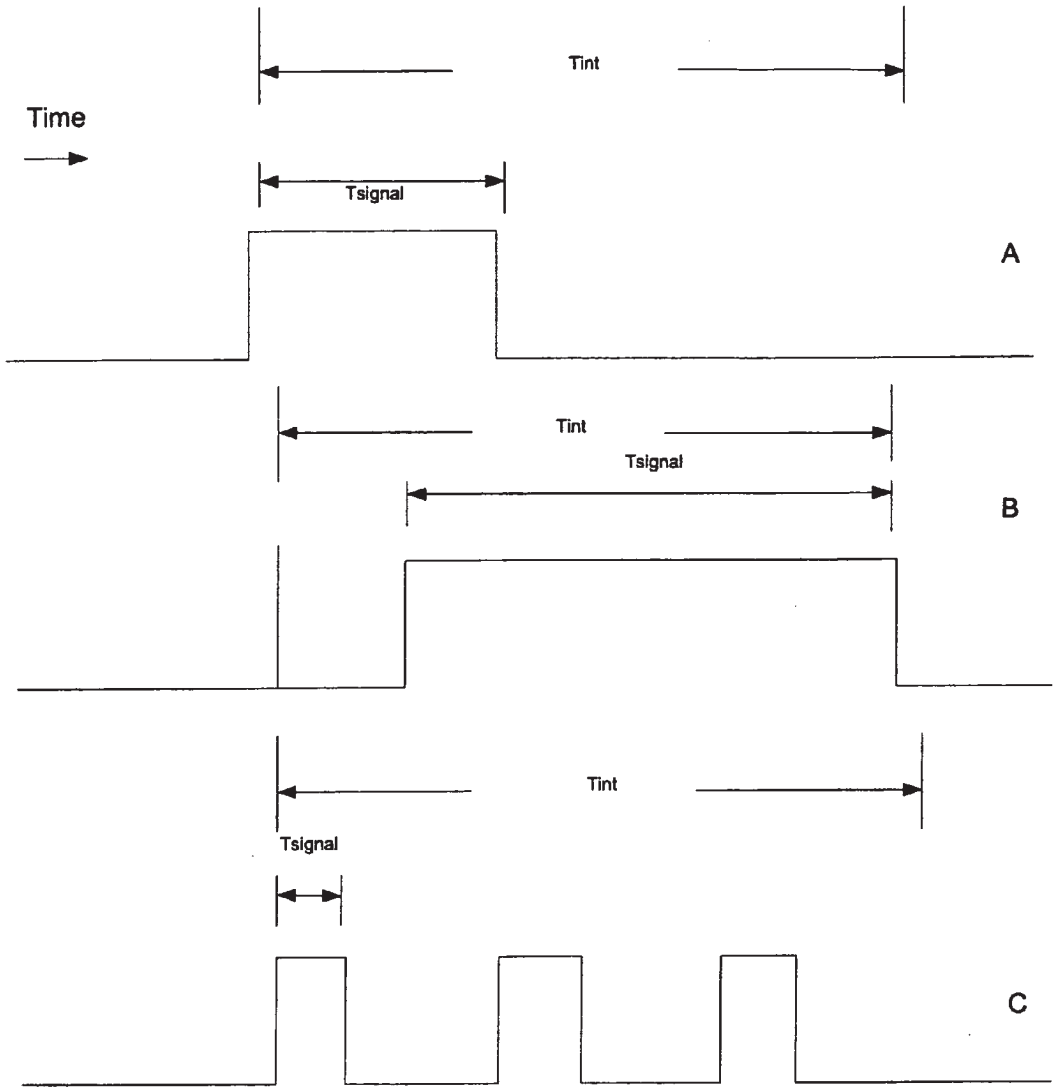


Fig. 29

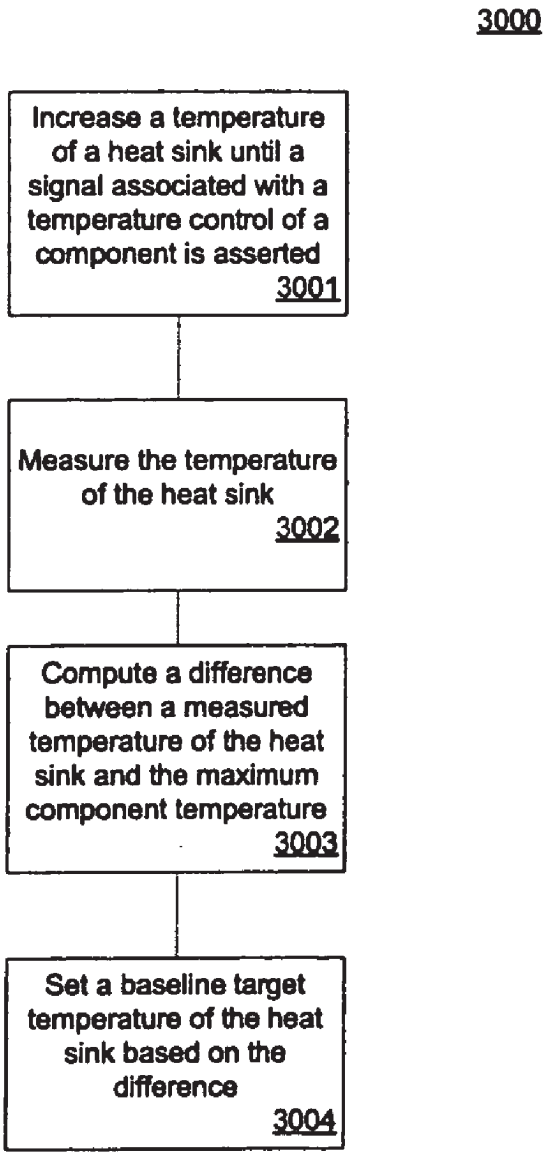

3000
Increase a temperature of a heat sink until a signal associated with a temperature control of a component is asserted
3001
Measure the temperature of the heat sink
3002
Compute a difference between a measured temperature of the heat sink and the maximum component temperature
3003
Set a baseline target temperature of the heat sink based on the difference
3004

Fig. 30

3100

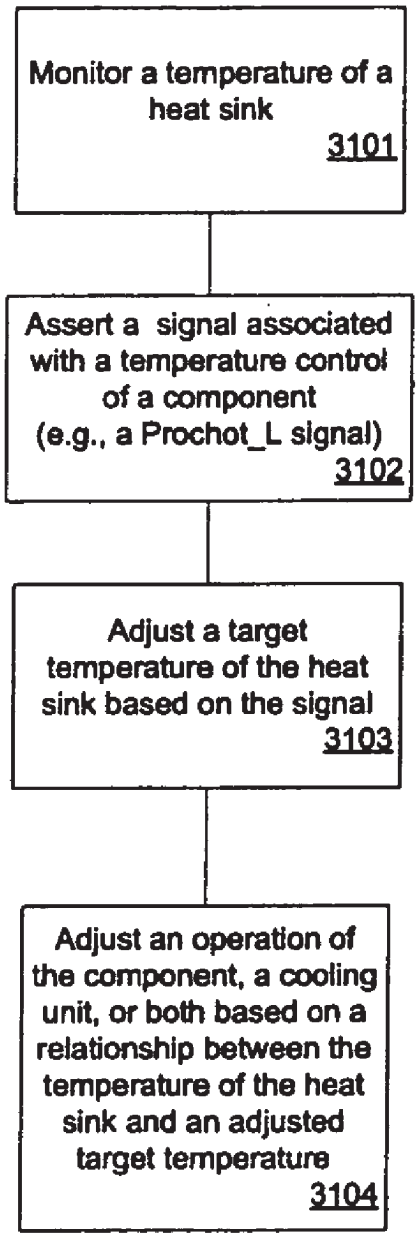

Monitor a temperature of a heat sink
3101
Assert a signal associated with a temperature control of a component (e.g., a Prochot_L signal)
3102
Adjust a target temperature of the heat sink based on the signal
3103
Adjust an operation of the component, a cooling unit, or both based on a relationship between the temperature of the heat sink and an adjusted target temperature
3104

Fig. 31

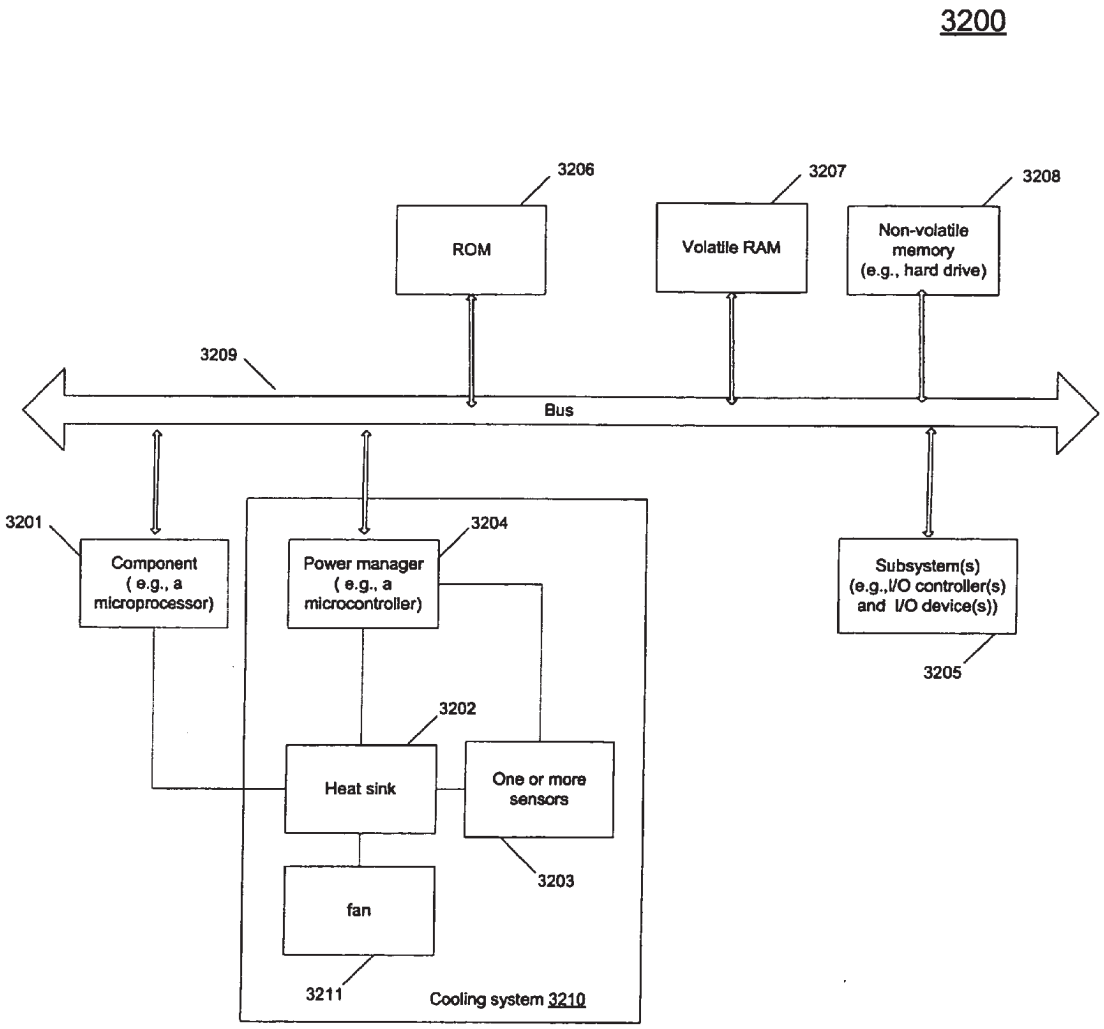

3200
3206
ROM
3207
Volatile RAM
3208
Non-volatile memory (e.g., hard drive)
3209
Bus
3201
Component (e.g., a microprocessor)
3204
Power manager (e.g., a microcontroller)
Subsystem(s) (e.g.,I/O controller(s) and I/O device(s))
3205
3202
Heat sink
One or more sensors
3203
fan
3211
Cooling system 3210

Fig. 32

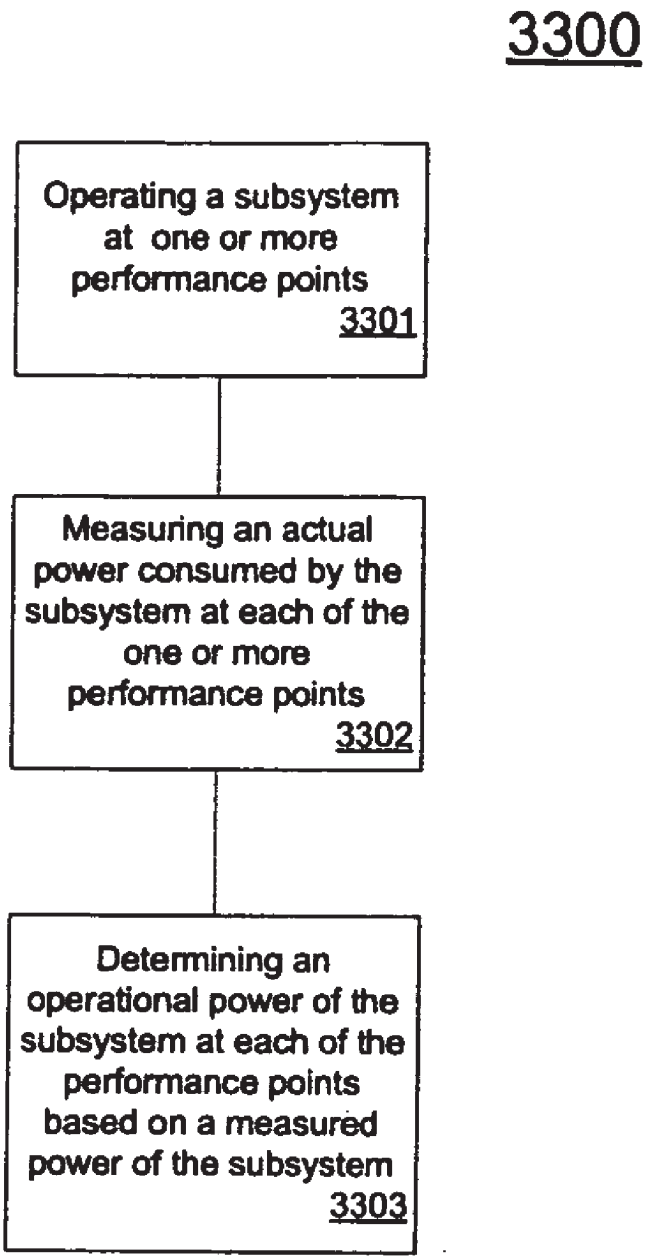

3300
Operating a subsystem at one or more performance points
3301
Measuring an actual power consumed by the subsystem at each of the one or more performance points
3302
Determining an operational power of the subsystem at each of the performance points based on a measured power of the subsystem
3303

FIG. 33

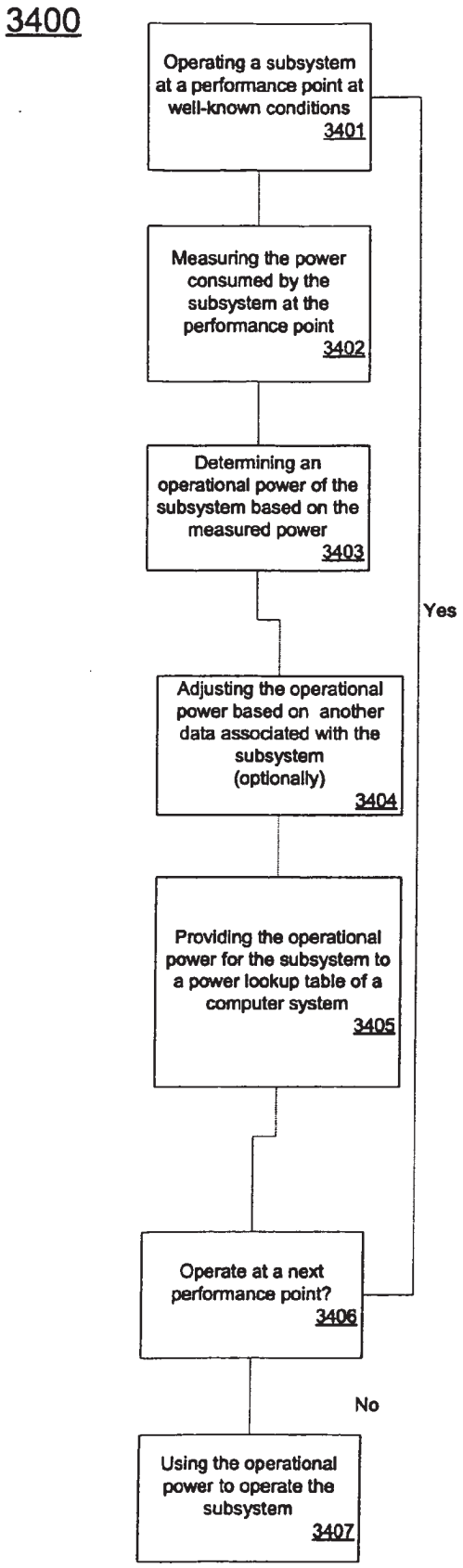

3400
Operating a subsystem at a performance point at well-known conditions
3401
Measuring the power consumed by the subsystem at the performance point
3402
Determining an operational power of the subsystem based on the measured power
3403
Adjusting the operational power based on another data associated with the subsystem (optionally)
3404
Providing the operational power for the subsystem to a power lookup table of a computer system
3405
Operate at a next performance point?
3406
Yes
No
Using the operational power to operate the subsystem
3407

FIG. 34

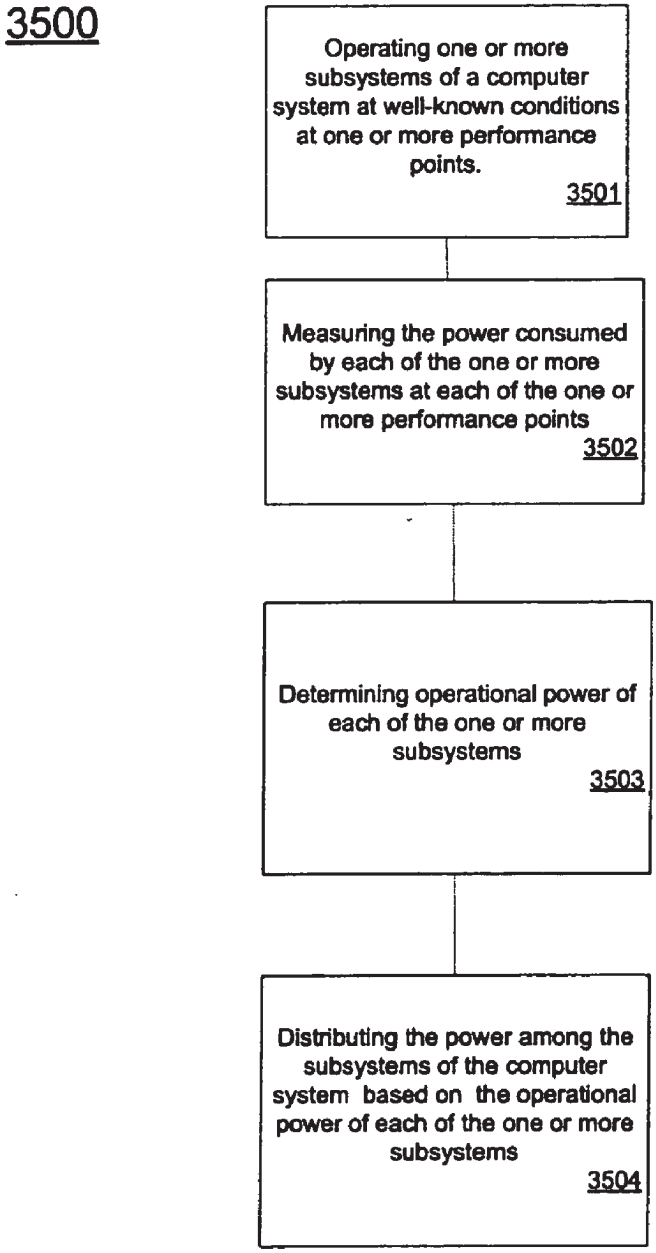

3500
Operating one or more subsystems of a computer system at well-known conditions at one or more performance points.
3501
Measuring the power consumed by each of the one or more subsystems at each of the one or more performance points
3502
Determining operational power of each of the one or more subsystems
3503
Distributing the power among the subsystems of the computer system based on the operational power of each of the one or more subsystems
3504

FIG. 35

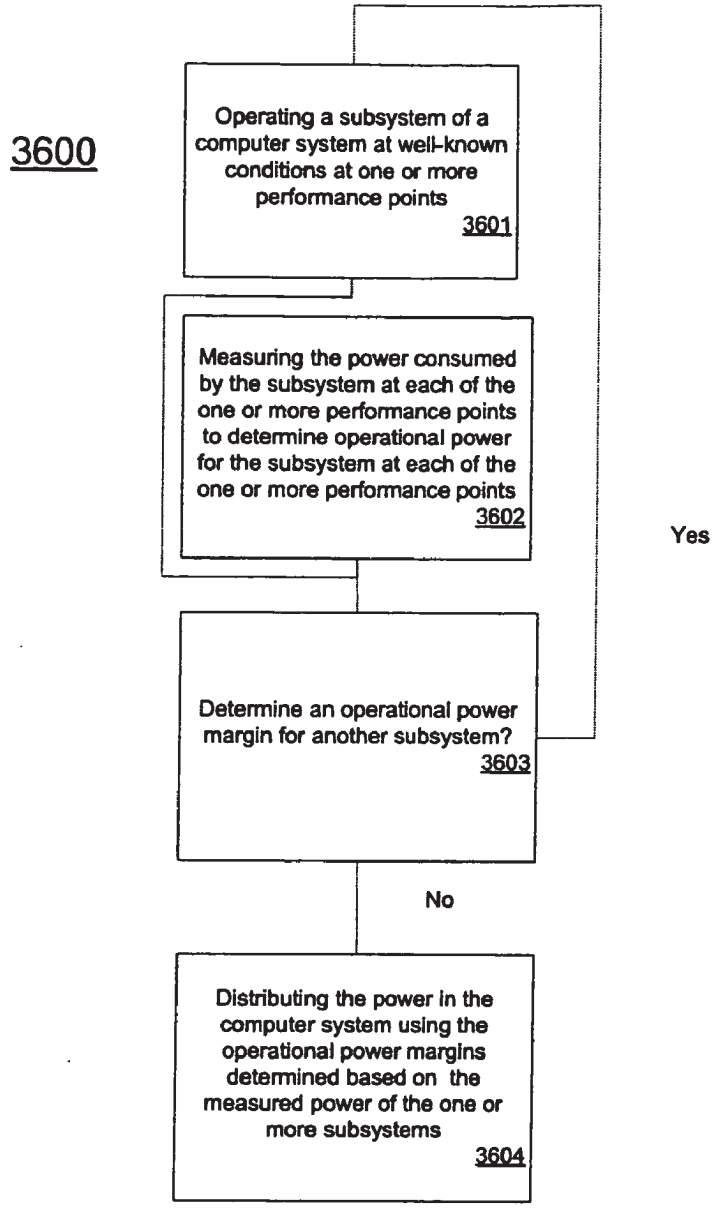

3600
Operating a subsystem of a computer system at well-known conditions at one or more performance points
3601
Measuring the power consumed by the subsystem at each of the one or more performance points to determine operational power for the subsystem at each of the one or more performance points
3602
Yes
Determine an operational power margin for another subsystem?
3603
No
Distributing the power in the computer system using the operational power margins determined based on the measured power of the one or more subsystems
3604

FIG. 36

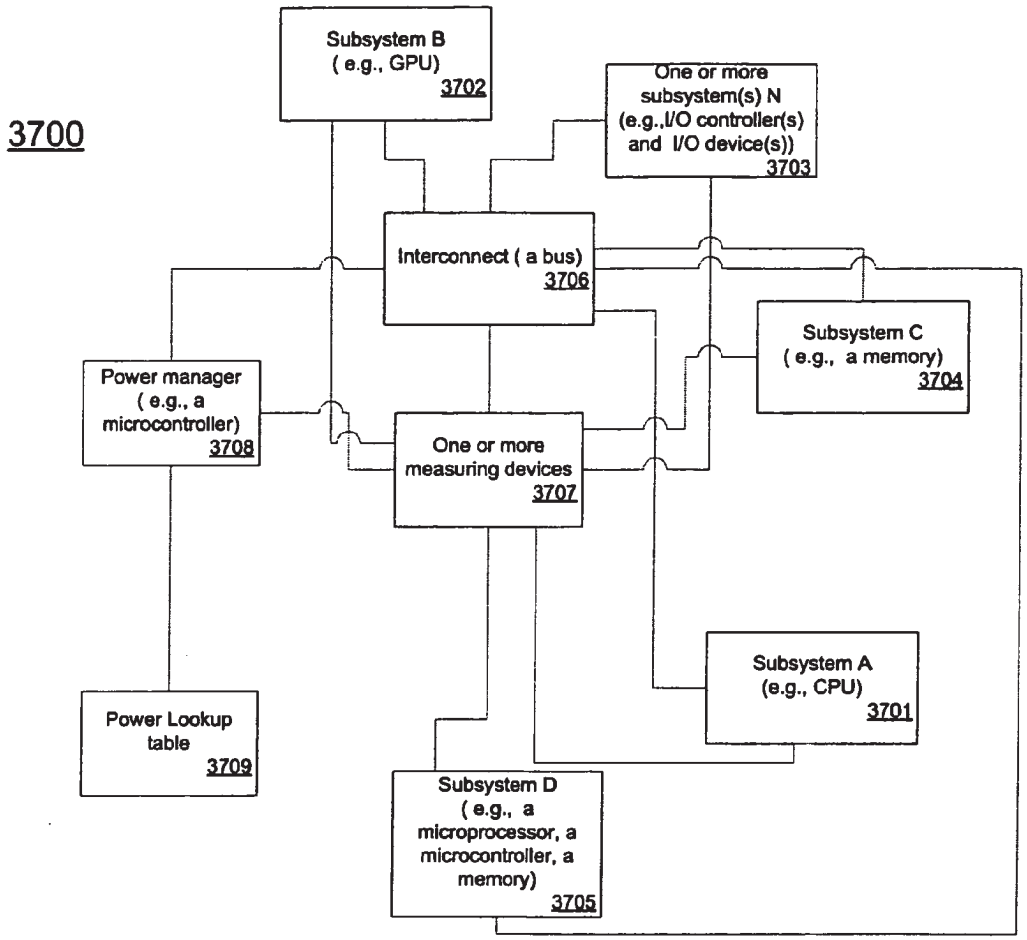

3700
Subsystem B
(e.g., GPU)
3702
One or more
subsystem(s) N
(e.g.,I/O controller(s)
and I/O device(s))
3703
Interconnect (a bus)
3706
Subsystem C
(e.g., a memory)
3704
Power manager
(e.g., a
microcontroller)
3708
One or more
measuring devices
3707
Subsystem A
(e.g., CPU)
3701
Power Lookup
table
3709
Subsystem D
(e.g., a
microprocessor, a
microcontroller, a
memory)
3705

FIG. 37

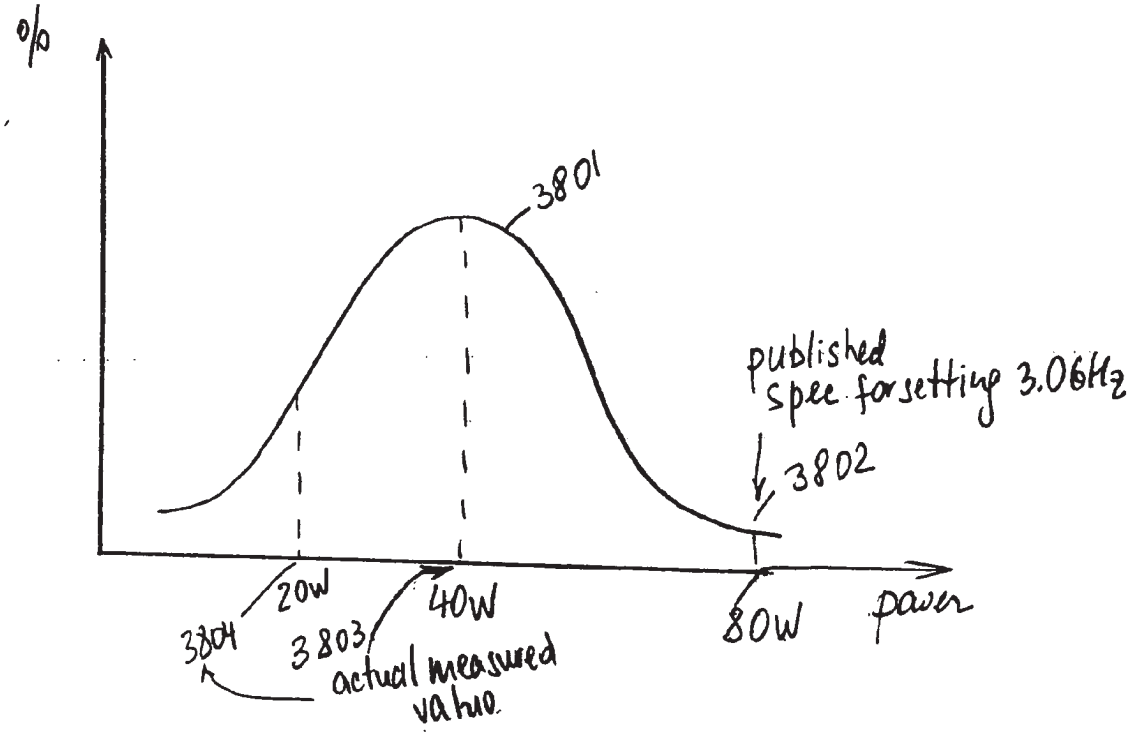

%
3801
published
spec. for setting 3.0GHz
3802
20W
40W
80W
power
3804
3803
actual measured
value

Figure 38.

METHODS AND APPARATUSES FOR DYNAMIC POWER CONTROL

RELATED APPLICATIONS

[0001] The present application is a Continuation-In-Part to U.S. patent application Ser. No. 11/212,970, filed on Aug. 25, 2005.

TECHNOLOGY FIELD

[0002] At least some embodiments of the present invention relate generally to data processing systems, and more particularly but not exclusively to the management of power usage in data processing systems.

BACKGROUND

[0003] Traditionally, computer systems are designed to be able to continuously run a fairly worst-case power load. Design according to such a continuous worst-case power load has never been much of a problem, because traditionally the individual components have had modest operating powers and the computer systems have had large power budgets so that the systems could sustain the load fairly naturally.

[0004] As the operating power consumptions of the individual components of computer system creep upwards, the power budgets of the computer systems have become tighter. It is now becoming a challenge to design a computer system to run a continuous worst-case workload while pursuing other high performance goals, such as high computing power, compactness, quietness, better battery performance, etc. For example, portable computer systems, such as laptop computers, have a limited battery output capability; and thus a worst-case workload for a given battery output capability may limit the performance of the system because the worst case workload may rarely occur.

SUMMARY OF THE DESCRIPTION

[0005] Exemplary embodiments of methods and apparatuses to dynamically redistribute power in a data processing system are described. The system includes a plurality of subsystems. The subsystems may be, for example, a central processing unit ("CPU") and a graphics processor ("GPU") of a computer system or other type of data processing system. The system that includes the plurality of the subsystems may be portable, such as a laptop computer.

[0006] In one embodiment, a load profile of the system is identified. The power is redistributed between the subsystems while tracking the load profile. The load profile may be an asymmetric, or a balanced load profile. For example, if in the system a workload of one subsystem is substantially higher than the workload of another subsystem, the load profile of the system may be asymmetric. That is, for an asymmetric load profile, the power of the system may be shifted toward the subsystem having a higher workload. Thus, higher power draw is allocated to and allowed for the subsystem having a higher workload. If the workloads of each of the subsystems are about the same, the load profile of the system is balanced. That is, for a balanced load profile, the power of the system may be distributed substantially evenly among the subsystems. In one embodiment, when all the subsystems in the system are in a low power operation, e.g., they are idling, the load profile of the system is balanced. In one embodiment, the load profile is identified based on a utilization factor for each of the subsystems. In one embodiment, to determine the utilization factor, the actual power used by each of the subsystems is measured. The actual power used by each of the subsystems may be measured using one or more sensors coupled to the subsystems. Next, the utilization factor is calculated based on the measured power. In one embodiment, the utilization factor is calculated as a ratio of the actual power used by the subsystem to the power allocated to the subsystem. Alternatively, the load profile may be calculated as a ratio of the actual power used by the subsystem to the power allocated to the system. Next, the load profile is determined using the utilization factor for each of the subsystems. A power weighting arrangement between the subsystems is selected based on the load profile. In one embodiment, the power weighting arrangement between the subsystems may be a power distribution table selected out of a plurality of power distribution tables stored in a memory of the system. The plurality of the power distribution tables include the power used by each of the subsystems at various system's settings.

[0007] In one embodiment, another subsystem is added to the plurality of the subsystems. Then, another load profile of the system that includes the another subsystem is identified. Another load profile may be identified using the utilization factor of each of the subsystems, including the another subsystem. Further, the power of the system is redistributed among the subsystems tracking another load profile.

[0008] One exemplary embodiment may perform as follows. Power distribution tables that correspond to various load profiles are generated. Each of the power distribution tables may include powers needed for each of the subsystems at various system power settings. An actual power used by each of the subsystems in the system is measured. A utilization factor for each of the subsystems in the system is determined. The utilization factor may be a ratio of the actual power used by the subsystem to the power budgeted ("allocated") to the subsystem. The power allocated to the subsystem may depend on a setting of the system. That is, the power allocated to the subsystem may be different for different settings of the subsystem. Further, a load profile of the system is determined using the utilization factor for each of the subsystems. In one embodiment, the load profile of the system may be calculated by subtracting the utilization factor of one subsystem from the utilization factor of another subsystem. Next, a power distribution table that corresponds to a current load profile is selected out of a plurality of distribution tables associated with various load profiles.

[0009] In one embodiment, the load profile of the system may be repeatedly, e.g., periodically, or continuously, identified. Then, the determination may be performed, whether a current load profile is different from the load profile determined previously. If the current load profile is different from the load profile determined previously, the power distribution table that corresponds to the current load profile is selected. In one embodiment, the plurality of the subsystems including, for example, a central processing unit, and a graphics processor perform asymmetric applications that result in asymmetric workloads for each of the subsystems.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings in which like references indicate similar elements.

[0011] FIG. **1** illustrates a method to dynamically control power usage according to one embodiment of the present invention.

[0012] FIG. **2** illustrates a method to dynamically determine power usage budget according to one embodiment of the present invention.

[0013] FIG. **3** illustrates a method to dynamically distribute power usage budget according to one embodiment of the present invention.

[0014] FIG. **4** illustrates a block diagram example of a system to dynamically control power usage according to one embodiment of the present invention.

[0015] FIG. **5** illustrates an example of dynamically throttling components of a data processing system to control power usage according to one embodiment of the present invention.

[0016] FIG. **6** illustrates an example of using throttle settings of a central processing unit (CPU) of a data processing system to control power usage according to one embodiment of the present invention.

[0017] FIG. **7** illustrates an example of using throttle settings of multiple components of a data processing system to control power usage according to one embodiment of the present invention.

[0018] FIG. **8** shows a method to dynamically determine throttle setting according to one embodiment of the present invention.

[0019] FIGS. **9-10** illustrate scenarios of power usage according to embodiments of the present invention.

[0020] FIG. **11** illustrates a table to look up the power usage requirement of the non-throttled component based on signal states according to one embodiment of the present invention.

[0021] FIG. **12** illustrates a computer system with a power management system according to one embodiment of the present invention.

[0022] FIGS. **13-16** illustrate methods of power management according to embodiments of the present invention.

[0023] FIG. **17** shows a block diagram example of a data processing system which may be used with the present invention.

[0024] FIG. **18** is a flowchart of one embodiment of a method to dynamically redistribute power in a system.

[0025] FIG. **19** is a flowchart of one embodiment of a method to dynamically redistribute power based on a load profile of a system.

[0026] FIG. **20** is a flowchart of another embodiment of a method to dynamically redistribute power based on a load profile of a system.

[0027] FIGS. **21**A-**21**C illustrate one embodiment of power distribution tables built for a system that includes a CPU and a GPU subsystems.

[0028] FIG. **22** illustrates one embodiment of one of the power distribution tables associated with a load profile Kn for a system, which includes a plurality of subsystems **1** to N.

[0029] FIG. **23** is a flowchart of one embodiment of a method to dynamically redistribute power while tracking a load profile of a system when another subsystem is added to the system.

[0030] FIG. **24** illustrates one embodiment of a system to dynamically redistribute the power while tracking a load profile of a system.

[0031] FIG. **25** is a flowchart of one embodiment of a method to adjust a target temperature of a computer system.

[0032] FIG. **26**A illustrates one embodiment of a system having a component, such as one or more microprocessors, coupled to a heat sink.

[0033] FIG. **26**B shows a model of heat dissipation from a component through a heat sink.

[0034] FIG. **27** is a flowchart of one embodiment of a method of operating an adaptive cooling control system of a computer system.

[0035] FIG. **28** is a flowchart of one embodiment of a method to adjust a target temperature of a heat sink based on a component-specific signal.

[0036] FIGS. **29**A-**29**C illustrate alternate embodiments of signals associated with the temperature of the component.

[0037] FIG. **30** is a flowchart of another embodiment of a method of using a component-specific signal in a cooling system that includes a heat sink.

[0038] FIG. **31** is a flowchart of one embodiment of a method to operate a cooling system that includes a heat sink.

[0039] FIG. **32** illustrates one embodiment of a computer system having an adaptive cooling arrangement.

[0040] FIG. **33** is a flowchart of one embodiment of a method to manage the power of a computer system that leverages intermediate power points.

[0041] FIG. **34** is a flowchart of one embodiment of a method of providing an intermediate processor power point for a subsystem.

[0042] FIG. **35** is a flowchart of another embodiment of a method of using intermediate operational power points to distribute power in a computer system.

[0043] FIG. **36** is a flowchart of one embodiment of a method of determining intermediate operational power points of one or more subsystems of a computer system.

[0044] FIG. **37** illustrates one embodiment of a system that leverages intermediate operational power points.

[0045] FIG. **38** illustrates one embodiment of an intermediate operational power for a subsystem at various performance points.

DETAILED DESCRIPTION

[0046] The following description and drawings are illustrative of the invention and are not to be construed as limiting the invention. Numerous specific details are described to provide a thorough understanding of the present invention. However, in certain instances, well known or conventional details are not described in order to avoid obscuring the description of the present invention. References to one or an embodiment in the present disclosure are not necessarily references to the same embodiment; and, such references mean at least one.

[0047] In one embodiment of the present invention, it is recognized that power consumptions in a computer system typically change frequently during typical usages. Typically, not all components are simultaneously in the maximum power consumption mode. Further, some components may not be in the maximum power consumption mode continuously for a long period of time. The power consumption of a component, such as the central processing unit (CPU) microprocessor, changes dramatically over time in typical usages. For example, in the prior art, the power supplies or a battery pack of computer system were designed to produce enough power to satisfy a worst case scenario in which all of the components of the computer system were drawing their maximum (in this case, worst level) amount of power. This worst case scenario essentially adds together the worst case, maximum level, power consumption; and the design takes this scenario into account when selecting a power supply or a battery pack. Thus, designing a computer system to sustain the worst-case load can be overly conservative for typical usages.

[0048] In one embodiment of the present invention, a computer system is designed to optimize various design goals for typical usages. However, worst-case load cases can occur. To avoid damage to the computer system, a dynamic power management system is used to dynamically budget the power usage of at least some of the components of the computer system such that, when the heavy tasks are imposed on the system, the system can trade performance for power consumption to stay within the power usage limit.

[0049] FIG. **1** illustrates a method to dynamically control power usage according to one embodiment of the present invention.

[0050] In FIG. **1**, a computer system has a number of different throttle settings. For example, the CPU processor of the computer system may be set to run under different core voltages and/or different core frequencies such that the system can be fully functional but at different performance levels to trade power usage and computing performance.

[0051] Typically, different throttle settings correspond to different performance levels. Further, different settings have different power requirements (e.g., **121**, **123**, **125**, . . . , **129**).

[0052] The power requirement at each setting is the maximum possible power usage at the corresponding setting. However, depending on the tasks being performed, the actual power consumptions can vary within a range.

[0053] For example, at setting X, the power usage range (**129**) is between the minimum possible power consumption (**109**) (e.g., when the system is in idle) and the maximum power consumption (**107**) (e.g., when the system is fully busy).

[0054] In one embodiment of the present invention, the system is so designed that the limit (**103**) for time averaged power usage is less than the power requirement (**107**) for setting X. Thus, if the system were allowed to be fully busy for a long period of time, the system might be damaged. However, it is recognized that in a typical usage the average of the dynamic power usage range (**129**) may be lower than the limit (**103**) so that the system can be in the setting X for a long period of time for high performance.

[0055] In one embodiment of the present invention, when the system receives a task that requires the system to be fully busy for a long period of time, the system automatically switches to a different setting (e.g., setting C) such that the limit (**103**) is not violated.

[0056] One embodiment of the present invention dynamically budgets the power usage for components of a data processing system, which may have a power usage constraint, such as thin-and-light portables computers, or larger portables, and/or small consumer desktops. For example, the constraint of heat dissipation on a computer system with a compact housing may limit the power usage. For example, to maintain the performance of a battery pack, there is a limit on average battery discharge current. Although the battery may allow a much higher instantaneous discharge current, the battery may have a much lower limit on average discharge current for a period of time to prevent significantly degrading the performance of the battery, or causing irreversible damage to the battery.

[0057] In one embodiment of the present invention, computer systems (e.g., portable computers or desktop computers) are designed for typical usages and may not sustain a continuous worst-case power load for a long period of time. Typically, a burst of computer tasks may require an instantaneous peak power load, which lasts only for a short period of time before the tasks are completed. Before and after the burst of the tasks, the computer system typically consumes a power load that is much lower than the worst-case power load. Since certain power usage limits are not based on the instantaneous power usage (e.g., the power limits based on thermal/heat dissipation constraint, based sustainable battery discharge current, etc.), such a temporary burst of tasks may be tolerable to allow high performance under tight power budget for typically usages.

[0058] Thus, with at least certain embodiments of the invention, performance levels (and power consumption levels) are set, for a short period of time (e.g., burst activity periods), to exceed a continuous state power capacity of the system (e.g., a power supply, or a battery pack). Traditionally, a computer system is designed according to the continuous worst-case workload; thus, no power usage range is allowed to exceed a limit for maximum continuous usage of a power supply (e.g., the capacity of a power supply to sustain a continuous state of constant power usage). For example, in a traditional design, no power usage range would exceed the limit (**103**), since the worst-case workload is traditionally treated as a continuous constant load. In one embodiment of the present invention, one or more power usage ranges (e.g., **129**) is allowed to exceed the limit (**103**) for a limited continuous period of time. In one embodiment of the present invention, the power usage of the system is managed such that the actual power usage is in average (e.g., over a time window based on a time constant of the power suppler) below the limit (**103**).

[0059] Although the power stack-ups which happen under worst-case power load can happen, they rarely do happen. Far more often one encounters a situation where some parts of the computer system are operating at full power, and other parts of the system are operating at low power. For example, when one is performing a scientific computation, the processor and the memory are very busy and are consuming close to full power, but the optical disk drive is empty and is consuming close to no power. When one is playing a DVD, the optical disk drive is consuming close to full power because it is reading the DVD, and the display is also consuming close to full power because it is very bright, but the processor and memory are consuming only modest power because decoding a DVD is not a very demanding application.

[0060] In one embodiment of the present invention, the power of a computer system is redistributed and throttled dynamically to ensure that the actual power usage is within the power usage constraints.

[0061] In one embodiment of the present invention, dynamic power redistribution is used to design a computer system for a typical power load over some averaging interval, as opposed to designing it for a continuous worst-case power load. Because a typical power load, in an average sense, is less than a worst-case power load, the system designed in this way can be constructed from higher-power components (which may be cheaper, or may have higher performance), or can be made physically smaller.

[0062] Because worst-case power stack-ups rarely do happen, a system designed in this fashion performs as well as a system designed in the traditional way in all but the most pathological situations. Unfortunately, worst-case power stack-ups can happen. Since a system designed for typical power cannot naturally handle worst-case conditions, one embodiment of the present invention uses throttling to ensure that the power constraints are not violated.

[0063] For example, in FIG. **1**, the system dynamically determine the "best" allowable setting according to the actual past power usage information to ensure that even if the computer is fully busy in the next time interval at the selected setting, the limit (**103**) will not be violated.

[0064] In one embodiment of the present invention, the system implements controls (throttles) on a subset of its subsystems which limit the maximum power that could be consumed by those subsystems. Adjusting the throttle settings can adjust the maximum power a subsystem (component) can use. Typically, these throttles limit the performance of the subsystem. For example, different throttle settings may be designed for different performance levels of the subsystem; and the power usage is limited as a side effect.

[0065] For example, the CPU (central processing unit) microprocessor may work using different core voltages and core frequencies. Using a high core voltage and a high frequency, the microprocessor can be optimized for computational performance but it has high power consumption. Using a low core voltage and a low frequency, the microprocessor can be optimized for battery performance at a degraded computational performance level.

[0066] In one embodiment, the microprocessor can shift from one core voltage level to another and from one frequency to another through slowly changing the voltage and through slowly change the frequency, without stopping or pausing the computational tasks of the processor. Typically, the voltage and the frequency are changed separately. For example, to move from a high frequency and a high core voltage to a low frequency and a low core voltage, the system changes the frequency while at the high voltage and then changes the voltage while at the low frequency. For example, to move from a low frequency and a low core voltage to a high frequency and a high core voltage, the system changes the voltage while at the low frequency and then changes the frequency while at the high voltage. Further details on changing CPU core voltage and frequency can be found in U.S. patent application Ser. No. 10/917,719, filed Aug. 12, 2004, which is hereby incorporated herein by reference.

[0067] In one embodiment of the present invention, the system dynamically redistributes power and sets these throttles so that even when the worst-case power load for these throttles occurs, the maximum power consumed over the averaging interval does not exceed the limit. Since worst-case power loads are rare, the throttle controls can normally be set to very high values, such that the system acts as if the limit does not exist when the demand of the system is lower than the dynamically determined budget.

[0068] FIG. **2** illustrates a method to dynamically determine power usage budget according to one embodiment of the present invention.

[0069] In FIG. **2**, actual power usages are monitored. For example, the actual power usage can be measured periodically to determine the history of the power usage. The history of the power usage can be used to determine the power usage in certain averaged ways.

[0070] In one embodiment of the present invention, with the knowledge of the past power usage (e.g., measurements **211**, **213**, . . . , **219** at times $T-(n-1)\Delta$, $T-(n-2)\Delta$, . . . , T) the system can dynamically determine the allowable power budget for the next time interval (e.g., for time $T+\Delta$).

[0071] For example, the limit (**203**) may be a simple average in a time window (**221**) of a time period $(n+1)\Delta$ (e.g., less than 10 seconds). Thus, in the example of FIG. **2**, the actual past power usage (e.g., **211**, **213**, . . . , **219**) can be used to determine the power budget (**205**) such that the area (**207**) of the power budget that is above the limit (**203**) is equal to or less than the area (**209**) between the limit (**203**) for average power usage (**203**) and the curve of the past power usage.

[0072] The system is then throttled to a setting that will not require power usage more than the dynamically determined budget (**205**).

[0073] In the next time period (e.g., $T+\Delta$), the actual power usage is measured, which is typically smaller than the power budget (**205**). Using the newly measured power usage information and the time window that advances with time for computing the average, the power budget and throttle setting for a further time period can be determined in a similar way.

[0074] In one embodiment, the power budget (**205**) is further limited according to other conditions, such as the limit (**201**) for instantaneous power usage.

[0075] Further, there may be a number of different types of average-based power usages (e.g., with different weight in averaging, based on different power limitation considerations, etc.) Thus, multiple copies of the power budget can be determined from a set of different computations, based on past power usage information; and the lowest power budget can be selected as the limiting budget that the determines the throttle setting.

[0076] In one embodiment, the measurement is an instantaneous data sample; and the frequency of the data samples are such that the changing history of the power usage is captured in a reasonably accurate way. Alternatively, the measurements may be obtained through an integration process such that even a small number of data samples can represent the power usage history with sufficient accuracy.

[0077] Note that the data samples may or may not be collected at the same time interval as that for computing the power budget and throttle setting. In one embodiment, the time period for determining the throttle setting is sufficiently short in comparison with the window (**221**) to allow flexibility in budgeting and long enough to provide sufficient time for the system to transit from one throttle setting to another when necessary and work reliably in the selected throttle setting.

[0078] The time period for determining the throttle setting may or may not be constant. For example, when a significant power usage event occurs (e.g., when the requirement estimate of the non-throttled components changes), the system may automatically start a new computation of the throttle setting.

[0079] FIG. **3** illustrates a method to dynamically distribute power usage budget according to one embodiment of the present invention.

[0080] In FIG. **3**, the dynamically determined power budget (**301**) is to be allocated to different components (subsystems) of the system.

[0081] In one embodiment, the system includes throttled component(s) and non-throttled component(s). A throttled component has different throttle settings at which the component is functional but at different power/performance levels (operating setting). For example, a processor may be throttled to work at different core voltages and core frequencies; a disk drive may be throttled to work at different spin rate; a bus may be throttled at different frequencies; etc. If a component is not throttled to trade performance for power usage, the component is considered a non-throttled component.

[0082] In FIG. **3**, the power budget (**301**) includes a throttled portion (**303**) and a non-throttled portion (**305**). The non-throttled portion corresponds to the estimated power used by non-throttled component(s) (**307**). The throttled portion is determined from the difference between the dynamically determined power budget (**301**) and the estimated power used by the non-throttled component(s).

[0083] In general, there can be one or more throttled components. When there are multiple throttled components, the throttle settings determined for the throttled components are such that the sum of the maximum powers (e.g., **311**, **313**, . . . , **319**) that can be used by the corresponding throttled components is no more than the throttled portion of the budget. Thus, the maximum powers (e.g., **311**, **313**, . . . , **319**) that can be used by the corresponding throttled components can be considered as budgets for the throttled components; and the throttle settings ensure that the actual powers (e.g., **321**, **323**, . . . , **329**) used by the throttled components are no more than their dynamically determined budgets (e.g., **311**, **313**, . . . , **319**).

[0084] Typically, the components (subsystems) whose throttle is adjusted may not actually consume the entire amount of power that is budgeted, since these components may not be busy enough to run at the maximum power corresponding to the throttle setting. The adjustment of the throttle allows the subsystem the freedom to consume up to the worse case power load for the corresponding throttle setting without violating power constraints.

[0085] When a worst-case power load appears, the system quickly notices the need for dynamic power redistribution and sets the throttles to lower values, keeping the system within its operating limits. In general, the power redistribution may be in the form of redistributing among subsystems (components) and/or redistributing over the time for the same subsystem (component) among different performance levels.

[0086] Imagine a system with an empty DVD drive that is running a scientific application. The processor and memory subsystems of the system are consuming close to full power, making the whole system run close to its operating limits. Now imagine that a disk is loaded into the DVD drive, which means that the DVD drive is to be enabled, consuming considerable power. In order to actually have power to enable the DVD drive, the system adjusts its power budget so that some of the power which used to be allocated to the processor is now allocated to the DVD drive; the throttle associated with the processor is switched to a lower value.

[0087] In one embodiment of the present invention, the averaging interval is (relatively) long with respect to the rate at which the dynamic power redistribution is executed. The allows the system to notice that the system is close to exceeding its limits, and have time to adjust the throttles and ensure that the system does not actually exceed its limits. The typical parts of the system which have power limits (batteries, heat sinks) tend to have fairly long time constants. Thus, it is easy to select a rate at which the dynamic power redistribution is executed.

[0088] Although power is referred to as an example of embodiments of the presentation inventions, other parameters related to power can also be managed in a similar way. For example, battery discharge current can be managed in a similar way as power.

[0089] In one embodiment of the present invention, a system with dynamic power management according to embodiments of the present invention includes one or more components (subsystems) that can be throttled to have different power requirements at different performance levels and have one or more sensors to actually determine the power consumed.

[0090] In one embodiment of the present invention, the past history of actual power consumption is used to dynamically determine the power usage budget for the subsequent time interval, such that even if the worst-case load occurs in

the subsequent time interval the power usage constraint (e.g., average power usage, or average battery discharge current) is not violated.

[0091] In one embodiment of the present invention, the actual power consumed by each subsystem is determined for the dynamic power redistribution and throttling.

[0092] In one embodiment of the present invention, instead of determining the actual power consumption by each subsystem, the sum of the power consumed by the throttled subsystems and the sum of the power consumed by the non-throttled subsystems are determined and used for the throttling.

[0093] FIG. **4** illustrates a block diagram example of a system to dynamically control power usage according to one embodiment of the present invention.

[0094] In FIG. **4**, a microcontroller (**411**) is used to budget the power usage dynamically. The power supply (**401**) (e.g., battery, AC adapter, etc.) provides power to the throttled component(s) (**409**) (e.g., CPU) and the non-throttled component(s) (**405**) (e.g., hard drive, DVD ROM, etc.). The microcontroller (**411**) can be considered as part of the non-throttled components. Alternatively, the microcontroller (**411**) may draw power from a power supply different from the power supply (**401**). Sensors (**407** and **403**) are used to determine the actual power usages by the throttled component(s) (**409**) and the non-throttled component(s). The microcontroller (**411**) collects the actual power usage information from sensors (**407** and **403**) and communicates with throttled components (**409**) to make throttle changes.

[0095] In one embodiment, a single sensor or measuring device may be used to measure power drawn by several non-throttled devices (rather than having one sensor for each non-throttled device). For example, wires may be arranged to connect to several non-throttled devices; and the measured power is that consumed by all of the non-throttled devices connected. A sensor can be used to determine the sum of the power consumed by the system directly (e.g., using a single current sensor at a location where the current drawn by the throttled components and the current drawn by the non-throttled components merges) and to determine the dynamic throttle setting.

[0096] Alternatively, this approach may be implemented by, for example, coupling the single sensor to wires from each of the several non-throttled devices, and the measured currents and/or voltages are summed in the sensor. Alternatively, multiple sensors can be used; and the microcontroller (or the microprocessor) sums the measurements from the sensors.

[0097] For example, the microcontroller may store the determined throttle setting in a register and then send a signal to the corresponding component (or the main CPU) to enforce the throttle setting. In one embodiment, the microcontroller sends the signal to enforce the throttle change only when the dynamically determined throttle setting is different from the previous one.

[0098] In one embodiment, the sensors are implemented using hardware. Alternatively, at least some of the sensors can be implemented using software. For example, software modules may be used to determine the operation states and corresponding time periods to compute the actual power usage from predetermined power consumption rate for the operation states.

[0099] FIG. **5** illustrates an example of dynamically throttling components of a data processing system to control power usage according to one embodiment of the present invention.

[0100] In FIG. **5**, the actual power usages include the power used by the throttled components (e.g., **521**, **511**, **513**, . . . **517**) and the power used by the non-throttled components (e.g., **523**, **512**, **514**, . . . **518**).

[0101] In one embodiment, the power used by the non-throttled components at the subsequent time interval (e.g., $T+\Delta$) is determined using the worst-case power load of the non-throttled component. Alternatively, the non-throttled components may be interrogated to obtain the worst-case power load of the non-throttled component according to their current operating status.

[0102] Alternatively, operating signals of at least some of the non-throttled components can be used to classify the corresponding non-throttled components into a global operating state, which is used to obtain an estimate that corresponds to the global operating state.

[0103] In one embodiment, the raw sensor measurements are used directly by a microcontroller or a microprocessor to perform dynamic power budgeting. Alternatively, the raw sensor measurements may be further processed using hardwire (e.g., using analog or digital circuitry) to generate data that is used by the microcontroller or microprocessor to perform dynamic power budgeting. Thus, there may be a layer of algebra between the raw sensors and the throttled and non-throttled powers.

[0104] In one embodiment, dynamic power redistribution is performed frequently and periodically. For example, the dynamically determined throttle setting can be determined periodically at a predetermined time interval. However, it is understood that the time interval is not necessarily a constant. For simplicity, some examples with a constant time interval are used to illustrate the methods according to embodiments of the present invention.

[0105] In one embodiment of the present invention, the setting for a throttle is computed for the next time interval to guarantee that the average power over the last N time intervals, including the next time interval, is less than a power constraint P_{MAX}. Thus, when the measured power data from the last N−1 time intervals is combined with the hypothetical worst-case power data of 1 sample for the next time interval, the average power is no more than P_{MAX}.

[0106] In general, the averaging process of the power usage over the last N time intervals can be a simple average, or a weighted average (e.g., weighted according to the elapsed time with decreasing weight for away back into the past), or other complex functions of the power usage history.

[0107] For example, let $P_T[N-1:1]$ be an array containing the measured power data of the throttled parts of the system for the last N−1 time interval. Let $P_N[N-1:1]$ be an array containing the measured power data of the non-throttled parts of the system. To determine the throttle for the next time interval:

[0108] 1) update the array containing the measured power data of the throttled parts of the system. For example, $P_T[N-2:1]$ can be copied (e.g., through shifting) into $P_T[N-1:2]$; and a new measurement of the sum of the power data of the throttled part of the system is stored into $P_T[1]$. The oldest sample, which was in $P_T[N-1]$, is discarded.

[0109] 2) similarly, update the array containing the measured power data of the non-throttled parts of the system. For example, $P_N[N-2:1]$ can be copied (e.g., through shifting) into $P_N[N-1:2]$; and a new measurement of the sum of the power data of the non-throttled part of the system is stored into $P_N[1]$. The oldest sample, which was in $P_N[N-1]$, is discarded.

[0110] 3) compute E_{PN}, which is an estimate of average value of the non-throttled power over the last N samples, from the average of the N−1 power measurements in $P_N[N-1:1]$ and one estimate of the maximum power $P_N[0]$ which may be consumed by the non-throttled parts of the system.

[0111] 4) for ith throttle setting, computer $E_{TN}[i]$, which is an estimate of the average value of the throttled power over the last N samples, from the average of the N−1 power measurements in $P_T[N-1:1]$ and the estimate of the maximum power $P_T[0, i]$ which may be consumed by the throttled parts of the system at throttle setting i.

[0112] 5) determine the highest throttle setting i_m for which E_{PN} $E_{TN}[i_m]$ is less than or equal to P_{MAX}. Thus, when throttle setting i_m is used, the average power through the next time interval will be less than or equal to the maximum power allowed by the system.

[0113] Note that in general, any throttle setting ix for which E_{PN} $E_{TN}[i_x]$ is less than or equal to P_{MAX} can be selected without exceeding the limit P_{MAX}. In one embodiment, the throttle settings are arranged according to performance level. The higher the throttle setting, the higher the performance. Thus, the highest throttle setting that limit the power usage according to P_{MAX} is selected to allow the highest performance under the power constraint.

[0114] FIG. **6** illustrates an example of using throttle settings of a central processing unit (CPU) of a data processing system to control power usage according to one embodiment of the present invention.

[0115] In FIG. **6**, a number of different combinations of CPU core voltages and core frequencies are sorted so that the throttle setting increases with the performance level, as illustrated in table **601**. In one embodiment, the system searches in the order of decreasing throttle setting to determine the first throttle setting that satisfies the relation E_{PN} $E_{TN}[i_x]<P_{MAX}$.

[0116] Thus, when a throttle setting is determined, both the CPU core voltages and frequencies are determined.

[0117] Alternatively, the throttles may be sorted according to other goals (e.g., a combined goal indicator to reflect the requirement for high computing power and low energy consumption, etc.); and a "best" throttle setting can be searched in a similar way.

[0118] Note that if there are multiple independent throttles, a list of different combination of throttles can be examined to determine the allowable throttle settings. A "best" setting of the throttles can be selected according to certain rules that define the objective "best". It is understood that the rules for define the objective can be arbitrarily complex.

[0119] FIG. **7** illustrates an example of using throttle settings of multiple components of a data processing system to control power usage according to one embodiment of the present invention.

[0120] In FIG. **7**, multiple components have independent throttle settings, as illustrated in table **701**. To distribute the dynamically determined budget to the multiple components, different combinations of the throttle settings for the multiple components can be viewed as different global throttle settings. The global throttle settings can be sorted according to a target goal level.

[0121] In one embodiment, the sorting of the global settings can be performed at the design stage of the computer according to a static fixed target goal function, or manually arranged by the designer of the system.

[0122] Alternatively, the global settings can be performed in real time according to a target goal function, which may be a function of current state of the computer system. For example, some of the components may be busy so that require higher priority while others may be in idle and require lower priority. Thus, the target function can be constructed to include the consideration of the current workloads of the components. The workloads can be estimated from the history of the actual power consumptions. For example, the high power consumption with respect to the dynamic power range of the component indicates a high workload for the component.

[0123] Once the global settings are sorted according to the target goal level, the highest global setting that satisfies the power constraint is selected.

[0124] FIG. **8** shows a method to dynamically determine throttle setting according to one embodiment of the present invention.

[0125] In FIG. **8**, the throttle settings are sorted according to the power requirements. Since the power requirements are typically known at the design stage, the sorting can be performed once during the design of the system.

[0126] The dynamically determined power usage limit (**301**) is partitioned into the non-throttled portion (**305**) and the throttled portion (**303**). The non-throttled portion (**305**) corresponds to the estimated power (**307**) used by the non-throttled components in the subsequent time interval.

[0127] The power budget (**811**) for the throttled components can then be used to determine the set of throttle settings (**813**) that are within the power budget limit (e.g., **801**, **803**, . . . , **805**). The throttle settings that are outside the power budget limit (**815**) will be excluded from consideration for the next time interval (e.g., **807**).

[0128] The system then can select one from the allowable set of throttle settings (**813**) to optimize a performance goal.

[0129] In one embodiment, when the previous actual power usage is low, the power budget (**811**) for the throttled component(s) can be sufficient enough to allow all throttle settings.

[0130] Typically, a selected throttle setting is used until the power measurement for the next time interval is obtained and the next iteration of dynamic throttling is performed.

[0131] Alternatively, the throttle setting may be determined on a substantially continuous basis; and the power management system requests throttle setting changes when necessary. To avoid frequent changes in throttle settings, the power management system may determine the throttle setting so that the throttle setting will be valid for at least a predetermined period of time unless a significant change in the estimate of the power used by the non-throttled components is detected (e.g., when a disk is loaded into the DVD ROM drive).

[0132] In one embodiment of the present invention, the power management monitors the actual power usage and adjusts the throttling to avoid the violation of power constraints.

[0133] FIGS. **9-10** illustrate scenarios of power usage according to embodiments of the present invention.

[0134] In the scenario of FIG. **9**, the computer system processes (**903**) low demand tasks before time instance (**911**); and the actual power (**921**) used by the system is below the limit (**915**) for average power usage. Since the power usage of the system is low, the system can be at the top performance setting (**901**).

[0135] After time instance (**911**), a high demand task is received. The system becomes busy in processing (**905**) the high demand task. Since the system had low power consumption before the reception of the high demand task, the power management allows the system to remain in the top performance setting for a short period of time. Thus, the actual power usage (**917**) increases to above the limit for average power usage. However, in average, the actual power usage is still below the limit.

[0136] After the high demand task is finished at time instance (**913**), the actual power usage (**923**) comes back to below the limit (**915**) for average power usage. Thus, the system can remain (**901**) in the top performance setting to process (**907**) low demand tasks.

[0137] The usage pattern as illustrated in FIG. **9** can be a typical one for certain usages of the system. Thus, the system can be designed on a tight power constraint while capable of running at top performance setting as if it were designed according to a worst-case load which would require a much higher power capacity.

[0138] However, the high demand task can be such that it may take a long period of time to finish the task. If the system were allowed to be in the top performance setting for a long period of time, the limit (**915**) for average power usage would be violated. A system according to embodiment of the present invention can automatically detect such situations and throttles accordingly to avoid exceeding the limit.

[0139] For example, in FIG. **10**, before time instance (**1021**) of receiving a high demand task, the system processes (**1011**) low demand tasks, as indicated by the low actual power usage (**1037**).

[0140] After the time instance (**1021**), the system processes (**1013**) the high demand task for a period of time.

[0141] Initially, the system remains in the top performance setting, which causes the actual power usage to be above the limit (**1041**) for average power usage. At the time instance (**1023**), the power management recognizes that the average of the actual power usage in the past period is approaching the limit (**1041**); and the system throttles into a reduced performance setting (**1003**).

[0142] At the reduced performance setting, the actual power consumption (**1033**) is below the limit (**1041**) for average power usage. Thus, at time instance (**1025**), the average power usage in the past may fall below the limit (**1041**) enough to allow the system to temporary back to the top performance setting (**1005**).

[0143] When the processing of the high demand task lasts for a long period of time, the system automatically switches between the top performance setting and the reduced performance setting periodically to have a long term average that is close to the limit (**1041**) for the average power usage.

[0144] Thus, under the control of the dynamic throttling system, the system processes the high demand task as fast as possible within the limit of power constraint.

[0145] In one embodiment of the present invention, multiple copies of throttle settings can be determined based on different constraints, for example, one for each boundary condition of power. The lowest one of the multiple copies of throttle settings is then used to ensure that the all constraints are satisfied. Typically, the performance is set by the subsystem which is most constrained.

[0146] The estimate of the maximum power which may be consumed by the non-throttled subsystems can be computed by a simple worst-case analysis (adding together the maximum values which could happen under any conditions) or by a more elaborate analysis based on the information provided by the subsystems and detailed knowledge of the subsystem's state.

[0147] The dynamic throttle setting determination can be performed in a variety of components in the computer system, including the main processor of the computer system, or a microcontroller dedicated to the dynamic power throttling task.

[0148] There are advantages to execute the dynamic budgeting in the main processor, such as reduced cost, and the elimination of any need to communicate between the main processor and whatever other agent that is alternatively used to perform the task. However, it is difficult to make an arrangement such that dynamic power management operates in all situations, including when the software in the main processor fails or is replaced with some other software which has no knowledge of the power management algorithm. Further, when the computer system is in an otherwise idling state, the periodic power management task may prevent the system from entering a low power state, or may periodically wake the system from the low power state.

[0149] When the ability to load throttle settings is reserved to the main processor of the computer system and the dynamic power throttle determination is not performed in the main processor, making the computed throttle setting the current throttle setting may become complicated. The throttle settings need to be communicated to the main processor; and in some situations, it may be necessary to

implement fail-safe mechanisms to deal with the (unlikely) case that the software running in the main processor ignores the command to load the throttles. The fail-safe mechanisms can be fairly crude, since they should only be engaged in emergency situations. For example, when the microcontroller determines that the average of the past N samples exceeds the power limit P_{MAX} for a number of continuous time intervals, the microcontroller may assume that the throttling settings are not enforced properly and automatically initiate a shutdown process.

[0150] In one embodiment, the estimation of the power usage is obtained from adding together the maximum powers which could be consumed by the non-throttled subsystems (components). Such an estimate can be done when the system is designed; and the result can be a constant. However, such an estimate is extraordinarily conservative, which may unnecessarily cause the system to force a throttled subsystem (component) into a low performance setting.

[0151] In one embodiment, the main processor performs a fairly detailed power analysis based on the characteristics of the subsystems and the current state of the subsystems. The analysis result is then used to determine the maximum power the non-throttled subsystems can consume at the current state of the operating conditions of the subsystems.

[0152] For example, the main processor may look at all of the system's USB (Universal Serial Bus) ports, and, if a device is actually plugged into the port, extract the descriptor from the device which reports the device's power consumption, and use the information from the descriptor in the power analysis.

[0153] Such a detailed analysis can result in best possible estimate. However, such a detailed analysis may require non-trivial changes to software running on the main processor to provide the power consumption information.

[0154] In one embodiment of the present invention, the signals used for normal operation of a subsystem (component) are used to determine the global state of the subsystem (component). The power requirement for the current global state is then used to determine the power requirement of the subsystem. Such an approach can generally improve the estimation of the power requirement of non-throttled components of the system without taking on the complexity of a detailed analysis, or making non-trivial changes to the software.

[0155] In one embodiment of the present invention, it is observed that many, if not most, of the non-throttled subsystems operate in one or more global states, and those states can be distinguished by looking at signals already necessary for the operation of the subsystem.

[0156] For example, a USB port is either in the empty state (where it consumes no power) or the in-use state (where is can consume as much as 2.5 W of power). These states are easily distinguished by looking at the enable signal on the USB power switch.

[0157] A USB port has a power switch which is enabled by software when a device is plugged in, and disabled by software when the device is unplugged. The power management can look at the digital enable for the power switch to learn if the connector is empty or full, which lets it decide if it should use 0.0 W or 2.5 W in the calculation.

[0158] Alternatively, a crude power measurement for the USB port can be used to determine whether or not the USB port is in the 0.0 W mode. Such a power measurement approached can be used in a system which does not enable/disable the switches.

[0159] Ports for an IEEE-1394 serial bus is like USB, except that the device connected to the IEEE-1394 port can draw more power than a USB device. In one embodiment, when the IEEE-1394 port is on, the power management budgets 8 W for the device.

[0160] A crude measurement of the disk drive current can tell if the disk is spinning or not (it can tell sleep state from idle state). If the disk is spinning, the maximum power that can be consumed by the disk drive is, for example, 2.5 W (e.g., for read/write access). If the disk is not spinning, it might be told to spin up, so the maximum power that can be consumed power is, for example, 5.0 W for a brief instant (e.g., for spin-up).

[0161] For example, a display backlight operates at a number of brightness levels, each with different power consumption; and these brightness levels are easily distinguished by looking at the PWM (Pulse-Width Modulation) control signal running between the display controller and the backlight power supply. A measurement of the duty factor of the PWM signal which runs between the graphics chip and the backlight inverter can be used to estimate the power consumed by the backlight. For example, a very dim backlight is less than a watt; and a full brightness backlight is close to 6 watts. Thus, improvement on the estimation of power used by the display backlight can significantly improve the estimation of power consumption by the non-throttled components.

[0162] For example, the PRSNT1#/PRSNT2# pin signals on the PCI (Peripheral Component Interconnect) slot (or similar pin signals from other interconnect slots, such as a PCI Express slot) could be used to estimate the power which might be consumed by the device plugged into the slot and to determine if a slot is empty.

[0163] In one embodiment, improved non-throttled power estimation is obtained from bringing the signals which allow the global power states to be distinguished to the agent (e.g., the microcontroller or the main processor) which actually needs the estimate. The power of any subsystem for which no state signal is available to determine the global states is assumed to need maximum power that could be consumed by the subsystem, but the power for any subsystem for which a state signal is available is assumed to be the maximum power in its current state.

[0164] Thus, no additional software modules are required to be running on the main processor for the purpose of reporting power requirements other than the software needed to actually run the device. Although the estimate it computes is no better than a worst case estimate in situations where all of the non-throttled subsystems are busy, it provides a considerably better than worst-case estimate in situations where some of the non-throttled subsystems are running at less than worst case. The inventors know that this is a very common case in normal operation. Thus, the

approach of using the existing operating signals in the estimation can provide a better estimate for typically usages.

[0165] FIG. **11** illustrates a table (**1101**) to look up the power usage requirement of the non-throttled component based on signal states according to one embodiment of the present invention. In one embodiment, the states of the signals are based on the existing signals are designed for the normal operations of the device. Thus, no special design or software module is necessary to obtain the power usage information from the device.

[0166] FIG. **12** illustrates a computer system with a power management system according to one embodiment of the present invention.

[0167] In FIG. **12**, interconnect (**1203**) connects various components (e.g., **1211**, **1213**, . . . , **1219**, **1221**, . . . , **1229**) with the main microprocessor(s) (**1201**).

[0168] In FIG. **12**, the power manager (**1207**) (e.g., implemented as a microcontroller) is used to dynamically determine the throttle settings of the system to balance the performance requirement and the power usage limit.

[0169] Signal sensors (**1205**) are used to monitor selected operating signals from some of the components (e.g., **1211**, **1213**, . . . , **1219**). These operating signals are used by the components (**1211**, **1213**, . . . , **1219**) even without the use of the power manager. The signal sensors (**1205**) tap into these signals to look up estimated power requirements for the components from lookup table (**1209**). A typical tapped signal can be used to classify the state of the component into one of a plurality of possible states. When operating in some of the states, the component consume less than the maximum possible power. Thus, the signals can be used to look up more accurate power requirements according to the operation condition of the components. Further, one of the sensors in the signal sensors (**1205**) may be measuring the power levels of one or several different components.

[0170] When a component does not have a signal tapped to obtain a better estimation, a conservative estimate based on the maximum possible power used by the component is used. The power manager adds the power requirement for the components (e.g., **1211**, **1213**, . . . , **1219**, **1221**, . . . , **1229**) to obtain an estimate of power that may be used in the subsequent time interval.

[0171] In one embodiment, based on the power requirement for these components and the past power usage history, the power manager (**1207**) further determines a throttle setting for the main microprocessor(s) (**1201**) so that the power usage within the next time period will be within the limit of a power constraint even when the main microprocessor(s) are fully busy in the next time period.

[0172] In one embodiment, the power manager is partially or entirely implemented as a software module running on the main microprocessor(s) (**1201**). The lookup table (**1209**) can also be implemented using a software module using a random access memory of the computer system or using a dedicated hardware module.

[0173] FIGS. **13**~**16** illustrate methods of power management according to embodiments of the present invention.

[0174] In FIG. **13**, operation **1301** determines actual power usage information during a first time period of operation of a data processing system. The actual power usage information can be in the form of measurement of power, or current (e.g., at a known voltage), or power averaged in time, or current averaged in time, or measurements of other quantities that are indicative of actual power usage. Operation **1303** determines a performance level setting of a component of the data processing system for a second time period subsequent to the first time period using the actual power usage information. The system is set to the determined performance level setting to ensure that the power usage of the system operating at the determined performance level setting in the second time period will not exceed any limit.

[0175] In FIG. **14**, operation **1401** obtains N−1 samples of actual power used during a time period T_1 of the operation of a data processing system which has a set of throttled components and a set of non-throttled components. In one embodiment, throttled components have different adjustable performance level settings that have different power usage requirements; and non-throttled components are not actively managed/controlled to trade performance level for power usage.

[0176] Operation **1403** estimates a power usage requirement of the non-throttled components in a subsequent time period T_2 of the operation of the data processing system.

[0177] Operation **1405** sorts different combinations of throttle settings according to a desired priority for processing in an order of decreasing priority. In one embodiment, the priorities of the throttle settings depend on the current workload of the different throttled components; and the sorting is performed in real time. In one embodiment, the priorities of the throttle settings are designed to be independent from the current workload of the different throttled components; and the sorting can be performed only once during the design or installation phase.

[0178] Operation **1407** processes one combination of throttle settings. Operation **1409** computes a power usage indicator based on the N−1 samples for time period T_1, the estimated power usage requirement of the non-throttled components for time period T_2, and the power usage requirement of the throttled components at the combination of throttle settings for time period T_2. For example, the power usage indicator can be an average power usage, an average battery discharge current, an average heat generation, etc.

[0179] If operation **1411** determines the computed power usage indicator is not allowable, operation **1413** processes the next combination; and operation **1409** is repeated, until operation **1411** determines the computed power usage indicator is allowable. In one embodiment, the settings are defined (e.g., by design) so that at least one combination is always allowable; thus, the loop of operations **1409**, **1411** and **1413** eventually exits to operation **1415**.

[0180] When the computed power usage indicator is allowable for a combination of throttle settings, operation **1415** selects this combination for the time period T_2. Operation **1417** throttles the throttled components according to the selected combination of throttle settings for the time period T_2.

[0181] Operation **1419** obtains one or more samples of actual power used during the time period T_2 while the throttled components are at the selected combination of

throttle settings. Operation **1421** shifts time window forward to determine throttle settings for the subsequent time period. Thus, operations **1403** through **1421** can be repeated for the subsequent time period.

[**0182**] In FIG. **15**, operation **1501** obtains one or more operating signals from a first component of the data processing system. In one embodiment, the operation signals are present for the normal operations of the components regardless whether or not the components are under power budget control according to embodiments of the present invention. Such an arrangement can minimize the impact of implementing methods of embodiments of the present invention on the design of the non-throttled components. Alternatively, the non-throttled components may be specifically designed to provide signals to dynamically indicate their power usage requirements.

[**0183**] Operation **1503** determines an estimate of a power consumption requirement for one or more components, including the first component, of the data processing system for operating under a current condition.

[**0184**] In FIG. **16**, operation **1601** obtains one or more operating signals from each of a first set of non-throttled components of a computer. Operation **1603** determines the global state of each of the first set of non-throttled components. Operation **1605** looks up a power usage requirement for each of the first set of non-throttled components according to the global state. Operation **1607** adds the power usage requirements of the first set of non-throttled components with the power usage requirement of the rest of non-throttled components to obtain the power usage requirement for the non-throttled components. Operation **1609** determines one or more throttle settings of a set of throttled components of the computer based on the actual power usage in the past and the power usage requirement for the non-throttled components.

[**0185**] Many of the methods of the present invention may be performed with a digital processing system, such as a conventional, general-purpose computer system. Special purpose computers, which are designed or programmed to perform only one function, may also be used.

[**0186**] FIG. **17** shows one example of a typical computer system which may be used with the present invention. Note that while FIG. **17** illustrates various components of a computer system, it is not intended to represent any particular architecture or manner of interconnecting the components as such details are not germane to the present invention. It will also be appreciated that network computers and other data processing systems which have fewer components or perhaps more components may also be used with the present invention. The computer system of FIG. **17** may, for example, be an Apple Macintosh computer.

[**0187**] As shown in FIG. **17**, the computer system **1701**, which is a form of a data processing system, includes a bus **1702** which is coupled to a microprocessor **1703** and a ROM **1707** and volatile RAM **1705** and a non-volatile memory **1706**. The microprocessor **1703**, which may be, for example, a G3, G4, or G5 microprocessor from Motorola, Inc. or IBM or a Pentium microprocessor from Intel is coupled to cache memory **1704** as shown in the example of FIG. **17**. The bus **1702** interconnects these various components together and also interconnects these components **1703**, **1707**, **1705**, and **1706** to a display controller and display device **1708** and to peripheral devices such as input/output (I/O) devices which may be mice, keyboards, modems, network interfaces, printers, scanners, video cameras and other devices which are well known in the art. Typically, the input/output devices **1710** are coupled to the system through input/output controllers **1709**. The volatile RAM **1705** is typically implemented as dynamic RAM (DRAM) which requires power continually in order to refresh or maintain the data in the memory. The non-volatile memory **1706** is typically a magnetic hard drive or a magnetic optical drive or an optical drive or a DVD RAM or other type of memory systems which maintain data even after power is removed from the system. Typically, the non-volatile memory will also be a random access memory although this is not required. While FIG. **17** shows that the non-volatile memory is a local device coupled directly to the rest of the components in the data processing system, it will be appreciated that the present invention may utilize a non-volatile memory which is remote from the system, such as a network storage device which is coupled to the data processing system through a network interface such as a modem or Ethernet interface. The bus **1702** may include one or more buses connected to each other through various bridges, controllers and/or adapters as is well known in the art. In one embodiment the I/O controller **1709** includes a USB (Universal Serial Bus) adapter for controlling USB peripherals, and/or an IEEE-1394 bus adapter for controlling IEEE-1394 peripherals.

[**0188**] In one embodiment of the present invention, at least some of the components can be actively throttled to trade performance for power usage. For example, the microprocessor **1703** may have different core voltage and frequency settings.

[**0189**] In one embodiment of the present invention, the system **1701** further includes power usages sensor(s) **1711** that are coupled to the I/O controller(s) **1709**. One or more sensors may be used to determine the power usage of the Central Processing Unit (CPU) (e.g., microprocessor **1703**) and/or the Graphical Processing Unit (GPU) (e.g., a processor of the display controller **1708**). Further, one or more sensor may be directly coupled to the CPU and/or GPU. The power usage sensor(s) **1711** may include one or more current sensors measuring the actual current drawn by the throttled components, and/or the actual current drawn by the throttled components, and/or the actual current drawn by the system. In one embodiment, the power usage sensor(s) **1711** may include a crude power usage sensor for a non-throttled component to determine the global state of the component, which can be used to dynamically estimate the power usage requirement of the component.

[**0190**] In one embodiment of the present invention, the microprocessor **1703** dynamically budgets power usage and determines throttle settings according to instruction stored in cache **1704**, ROM **1707**, RAM **1705**, and/or nonvolatile memory **1706**. Alternatively, the system **1701** further includes a microcontroller (not shown in FIG. **17**) to dynamically budget power usage and determine throttle settings. In one embodiment, the data processing system may include multiple central processing unit (CPU)/microprocessors.

[**0191**] It will be apparent from this description that aspects of the present invention may be embodied, at least

in part, in software. That is, the techniques may be carried out in a computer system or other data processing system in response to its processor, such as a microprocessor or a microcontroller, executing sequences of instructions contained in a memory, such as ROM **1707**, volatile RAM **1705**, non-volatile memory **1706**, cache **1704**, or other storage devices, or a remote storage device. In various embodiments, hardwired circuitry may be used in combination with software instructions to implement the present invention. Thus, the techniques are not limited to any specific combination of hardware circuitry and software nor to any particular source for the instructions executed by the data processing system. In addition, throughout this description, various functions and operations are described as being performed by or caused by software code to simplify description. However, those skilled in the art will recognize what is meant by such expressions is that the functions result from execution of the code by a processor, such as the microprocessor **1703**, or a microcontroller.

[**0192**] A machine readable medium can be used to store software and data which when executed by a data processing system causes the system to perform various methods of the present invention. This executable software and data may be stored in various places including for example ROM **1707**, volatile RAM **1705**, non-volatile memory **1706** and/or cache **1704** as shown in FIG. **17**. Portions of this software and/or data may be stored in any one of these storage devices.

[**0193**] Thus, a machine readable medium includes any mechanism that provides (i.e., stores and/or transmits) information in a form accessible by a machine (e.g., a computer, network device, personal digital assistant, manufacturing tool, any device with a set of one or more processors, etc.). For example, a machine readable medium includes recordable/non-recordable media (e.g., read only memory (ROM); random access memory (RAM); magnetic disk storage media; optical storage media; flash memory devices; etc.), as well as electrical, optical, acoustical or other forms of propagated signals (e.g., carrier waves, infrared signals, digital signals, etc.); etc.

[**0194**] The methods of the present invention can be implemented using dedicated hardware (e.g., using Field Programmable Gate Arrays, or Application Specific Integrated Circuit) or shared circuitry (e.g., microprocessors or microcontrollers under control of program instructions stored in a machine readable medium. The methods of the present invention can also be implemented as computer instructions for execution on a data processing system, such as system **1701** of FIG. **17**.

[**0195**] FIG. **18** is a flowchart of one embodiment of a method to dynamically redistribute power in a system. The method begins with operation **1801** of identifying a load profile of a system. The system has a plurality of subsystems. In one embodiment, the plurality of subsystems includes processors, e.g., a CPU, a GPU, a microcontroller, and the like. The power used by at least a subset of each of the subsystems is controlled, e.g., by a microcontroller, and a maximum power used by each of the subsystems is determined by the dynamic power history of the whole system over an averaging period, as described above with respect to FIGS. **1-17**. Such control of power allows higher performance operation in at least certain environments. That is, the subsystems may operate at bursts of substantially high power if there is a considerable low power operation, e.g., a idle time, during the averaging period, as described above with respect to FIGS. **1-17**. In one embodiment, the power of the subsystems may be controlled such a way that the maximum power used by each of the subsystems may be increased or reduced in unison, e.g., synchronously.

[**0196**] A load profile of the system is defined by workloads of each of the subsystems in the system. A workload of a subsystem may be determined using various techniques. In one embodiment, a workload of a subsystem determines the amount of power used by the subsystem in the system. In another embodiment, the operating system may determine the workload of the subsystem out from historical scheduling data, or an application may explicitly inform the system about the workload. Various applications provide various workloads to each of the subsystems. For example, program development tools and scientific applications present a high load to the CPU, but almost no load to the GPU that leads to an asymmetric load profile of the system (e.g. the CPU consumes a lot more power than the GPU). Many professional applications present an alternating high workload to the CPU and to the GPU that results in an alternating asymmetric load profile of the system. Advanced user interfaces or graphics editing application present a high load to the GPU and a modest load to the CPU that leads to another asymmetric load profile to the system. In one embodiment, the load profile may be identified using workloads determined by measuring/sensing power (e.g. current drawn) by each subsystem or by measuring power for certain subsystems and estimating or predicting power for other subsystems or by estimating power for all subsystems. In another embodiment, the load profile may be identified using workloads determined by the operating system out from historical scheduling data. In yet another embodiment, to identify the load profile of the system, the information about the workload of the subsystem provided by an application may be used.

[**0197**] After determining the load profile of the system, the method **1800** continues with operation **1802**, of redistributing the power of the system among the subsystems based on the load profile. In one embodiment, the power is redistributed in an asymmetric fashion, tracking the workloads of each of the subsystems. Such an asymmetric redistribution of the power improves the user experience, because it allows a system that is incapable of running all of its subsystems at a full speed to appear to be able to do so for many applications. In particular, detecting the asymmetric workloads of the subsystem, and redistributing the power in asymmetric fashion while tracking the workloads of the subsystems is important for small data processing systems such as portable computers or small desktop computers or handheld systems that may be incapable of running all of their subsystems at full speed.

[**0198**] FIG. **19** is a flowchart **1900** of one embodiment of a method to dynamically redistribute power by tracking a load profile of a system. The method begins with operation **1901** by sensing an actual power used by each of subsystems in a system. In one embodiment, a maximum power of a subsystem is controlled, as described above with respect to FIGS. **1-17**. In one embodiment, an actual power used by each of the subsystems is measured by one or more sensors (not shown) coupled to each of the subsystems. One or more sensors may be connected to each of the subsystems using,

for example, a wire, or the sensors may be directly attached to the subsystems. In one embodiment, the one or more sensors used to measure actual power usage by each of the subsystems are the sensors as described above with respect to FIGS. **4**, **5**, **12**, and **17**. Alternatively, power consumed by one or more subsystems may be estimated. Next, in operation **1902**, an utilization factor for each of the subsystems in the system is determined. The utilization factor may be considered as a power efficiency metric for a subsystem in the system. In one embodiment, the utilization factor is a ratio of the power used by the subsystem, e.g., the power measured by a sensor over a time interval divided by the power that is budgeted ("allocated") to the system (or the subsystem itself) over the same time interval. In one embodiment, the power for the system is allocated through the power redistribution algorithm, using the dynamic power history of the whole system over an averaging period, as described above with respect to FIGS. **1**-**17**. In one embodiment, the utilization factor for each of the subsystems is a number between 0 and 1.0. In alternate embodiments, other numbers for the utilization ratio for each of the subsystems may be used. A value of 1.0 may indicate a full utilization of allocated system power by a subsystem, and a value of 0 may indicate that the subsystem is in a low power mode, e.g., is turned off, idle, or in a sleeping mode.

[**0199**] Next, the method **1900** continues with operation **1903** which involves determining a load profile of the system based on the utilization factors of each of the subsystems. In one embodiment, each of the subsystems has a controlled operating power. In one embodiment, the load profile of the system is calculated using the utilization factors of each of the subsystems. For example, to calculate the load profile of the system having two subsystems, a utilization factor of one subsystem is subtracted from the utilization factor of the other subsystem. For example, in the system that contains two subsystems, if the utilization factor of a first subsystem is 0.25, and the utilization factor of a second subsystem is 0.75, the load profile of the system is 0.5. That is, the load profile is shifted asymmetrically towards the second subsystem. In one embodiment, the load profile of the system, which contains two subsystems calculated using the utilization factors of each of the subsystems that are numbers in the approximate range of 0.0 to 1.0, is a number in the approximate range of 1.0 to −1.0. Further, a value near 1.0 or −1.0 may indicate a substantially asymmetric load profile of the system. The substantially asymmetric load profile means that the power of the system could be shifted to operate one or more of the subsystems at high power, while the other one or more of the subsystems operate at a low power, e.g., are idle. Further, for example, if the utilization factor of the first subsystem and the utilization factor of the second subsystem are approximately equal, the load profile is about 0.0. Numbers about 0.0 may indicate that the load profile is balanced. That is, the power of the system could be evenly redistributed among subsystems in the system. In another embodiment, the load profile is an array of numbers that includes the utilization factors of each of the subsystems. For example, for the system having two subsystems, the load profile of [1.0, 0.5] or [−0.5, −1.0] is an asymmetric load profile, and the load profile of [0.5, −0.5] is a balanced load profile. It will be appreciated that a variety of alternative ways to define and calculate the load profile and/or the utilization factors may be used in alternative embodiments of the invention.

[**0200**] After determining the load profile of the system, the method **1900** continues with operation **1904** which involves selecting a power weighting strategy (which may be considered a power weighting arrangement among the subsystems) of the system based on the load profile. In one embodiment, the power weighting strategy is selected based on the value of the load profile. The power weighting strategy may be an asymmetric weighting, or a balanced weighting. For example, if the load profile is about 1, −1, [1.0, 0.5], or [−0.5, −1.0], a substantially asymmetric power weighting arrangement among the subsystems is selected. For example, if the load profile is about 0.0, [0.5, 0.5], or [0.5, −0.5], a substantially balanced power weighting arrangement among the subsystems is selected. In one embodiment, if all subsystems of the system are idle, the load profile is about 0.0, and the balanced strategy is selected. That is, the system is balanced at rest, and only moves towards an unbalanced strategy when it is actively unbalanced. In certain embodiments, this allows the system to correct itself very quickly by shifting from an asymmetric power strategy to a balanced power strategy.

[**0201**] In one embodiment, a power weighting strategy is selected based on the load profile by selecting a power distribution table out of a plurality of power distribution tables stored in a memory of the system. Generating the power distribution tables corresponding to various load profiles is described in further detail below with respect to FIGS. **22**A to **22**C, and FIG. **23**.

[**0202**] FIG. **20** is a flowchart **2000** of another embodiment of a method to dynamically redistribute power while tracking a load profile of a system. The method **2000** begins with operation **2001** of sensing (e.g. measuring) an actual power used by each of subsystems in a system, as described above with respect to FIG. **19**. Alternatively, power used by some subsystems may be measured while power used by other subsystems may be estimated or power used by all subsystems may be estimated. The method **2000** continues with operation **2002** which involves calculating for each of the subsystems a ratio of the power used by a subsystem to the power allocated to the system or that subsystem (the utilization factor). The utilization factor for each of the subsystems may be calculated as described above with respect to FIG. **19**. Next, in operation **2003**, a current load profile of the system is determined using ratios of each of the subsystems, as described above with respect to FIG. **19**. In operation **2004**, a determination is made whether the current load profile is different (or different enough) from a load profile determined in a time interval preceding the current time interval. In one embodiment, the current load profile of the system is compared with the previous load profile stored in a memory. If the current load profile is different from the previous load profile, a power distribution table corresponding to the current load profile is selected in operation **2005**. If the current load profile is not different (or not different enough, for example, the difference is less than a threshold value) from the previous load profile, the method continues with operation **2001** of sensing the power used by each of the subsystems.

[**0203**] FIGS. **21**A-**21**C illustrate one embodiment of power distribution tables built for a system that includes a CPU subsystem and a GPU subsystem. The power distribution tables depicted in FIGS. **21**A-**21**C correspond to different load profiles, e.g., asymmetric and balanced load

profiles. As shown in FIGS. **21**A-**21**C, the table **2100** has columns **2101**, **2102**, **2103**, and **2104**, and rows **2105**. Rows **2105** correspond to different settings of the system for a balanced load strategy. Settings of the system may be, e.g., different clock, and different core voltage settings, as described above with respect to FIGS. **1**-**17**. Each setting is associated with an amount of power P allocated to the system. As shown in FIGS. **21**A-**21**C, the amounts P**1**-P**3** of power P allocated to the system at settings **1**-**3** is the same for all tables **2100**, **2110** and **2120**. Column **2104** includes amounts P**1**-P**3** of power P allocated for the system at various system's settings. In one embodiment, allocating the power P corresponding to different system's settings within a given table is performed using methods described above with respect to FIGS. **1**-**17**. Columns **2102** and **2103** include amounts of power that each of the subsystems, e.g., CPU and GPU, uses to operate at different system settings. FIG. **21**A illustrates a power distribution table that corresponds to a balanced load profile K**1** (e.g., 0.5, 0.5). As shown in FIG. **21**A, each of CPU and GPU consumes about a half of the power allocated to the system at each of the settings **1**-**3**. That is, for table **2100** the power of the system is substantially evenly distributed (e.g. allocated or budgeted) among the subsystems and corresponds to a balanced load profile K**1** e.g., [0.5, 0.5] of the system. Table **2100** may be selected to run applications that require both CPU and GPU to be about equally busy. FIG. **21**B illustrates a power distribution table that corresponds to a CPU-heavy load profile K**2** e.g., [0.75, 0.25]. As shown in FIG. **21**B, at each of settings **1** to **3**, CPU consumes about 75%, while GPU consumes about 25% of the power allocated for the system. That is, for table **2110**, the power of the system is shifted towards CPU and corresponds to an asymmetric load profile K**2** e.g., [0.75, 0.25], of the system. Table **2110** may be selected when workload of the system is very CPU intensive, such that CPU consumes a substantially bigger share of the total system's power, while a graphics processor is hardly used at all. FIG. **21**C illustrates a power distribution table that corresponds to a GPU-heavy load profile K**3** e.g., [0.25, 0.75]. As shown in FIG. **21**C, at each of the settings **1** to **3**, GPU consumes about 75%, while CPU consumes about 25% of the power allocated for the system. That is, for table **2120**, the power of the system is shifted towards GPU and corresponds to an asymmetric load profile K**3** e.g., [0.75, 0.25] of the system. Table **2110** may be selected when workload of the system is very graphics intensive, such that GPU consumes a substantially bigger share of the total system's power, while a CPU is hardly used at all. As shown in FIGS. **21**A-**21**C, tables **2100**, **2110**, and **2120** are built such a way that if a system moves from one system setting, e.g. from setting **1**, to another system setting, e.g., setting **2**, the power needed to operate each of the subsystems increase or decrease at the same time. That is, for each of tables **2100**, **2110**, and **2120**, when performance of the system transitions between system power settings **1**-**3** within the same power distribution table, the load profile of the system does not change. Selecting a table out of a plurality of power distribution tables associated with different load profiles while tracking the workloads of each of the subsystems using one or more sensors provides an opportunity to select a proper table, e.g., table **2100**, **2110**, or **2120**, based on the load profile of the system at a current moment of time. That is, by just following a power usage pattern, the system's character can be dynamically changed with a substantially high accuracy, so as to be a balanced system, a CPU-heavy machine, a GPU-heavy system, or any other subsystem-heavy system, without using complicated software. As a result, efficient, dynamic power management for the system is provided, where portions of the hardware (processors, buses, memories, and other subsystems) may have their performance, and, as a side effect, their power, adjusted fairly transparently to the software, based on the present workloads of the subsystems. For example, the power of the system may be provided to a more busy GPU, while causing the CPU to slow down without affecting performance of the CPU. That is, a GPU can work at a higher speed in exchange for slowing down the operations on the CPU without visibly affecting the user's perception of the performance of the CPU while the GPU's performance is visibly improved. In another embodiment, for compute bound applications and memory bound applications, the power of the system may be dynamically redistributed between the CPU and a memory, while tracking actual workloads of each of the CPU and the memory. In yet another embodiment, the performance of a system, e.g., a portable computer, may alternate between GPU-heavy table **2120** and CPU-heavy table **2110** continuously (this kind of workload happens frequently in the frame-by-frame processing of digital video). This dynamic shifting of allocated power to various subsystems, based on monitoring a current load profile, is particularly useful for small, thin laptop computers which have small batteries and may not have cooling systems with large cooling capabilities, and this dynamic shifting is also particularly useful for small desktop computers (e.g. Mac Mini) which may not have cooling systems with large cooling capabilities.

[0204] FIG. **22** illustrates one embodiment of one of power distribution tables **2200** associated with a load profile Kn for a system, which includes a plurality of subsystems **1** to N. In one embodiment, subsystems **1** to N include core logic functions, memory, CPU, disk drives, GPU, peripheral devices, buses, and any other devices that have controlled power. As shown in FIG. **22**, the power distribution table includes columns **2201**-**2205**, and rows **2207**. Rows **2207** correspond to different settings of the system, as described above with respect to FIGS. **21**A-**21**C. Column **2206** includes powers allocated for the system at different system's settings, as described above with respect to one of the tables in FIGS. **21**A-**21**C. In one embodiment, amounts P**1**-Pm of power P may be placed in descending, or ascending order across column **2206**. Columns **2201**-**2204** include amounts P**1**-Pn of power P that each of the subsystems **1**-N is allocated to operate at different system settings. The system may have any number of power distribution tables **2200** associated with different load profiles of the system depending on the operating points that are needed to be enabled in a system. In one embodiment, the plurality of tables **2200** have the same amounts P**1**-Pm of power P allocated for the system at settings **1** . . . M. Tables, such as table **2200**, are created in such a way that a sum of the powers needed to operate each of the subsystems at each of settings **1**-M (across each of rows **2207**) does not exceed a corresponding amount of power P**1**-Pm allocated to the system at a corresponding setting **1**-M. Tables, such as table **2200**, may differ from one another by a proportion of work that each subsystem **1**-N is allowed to perform for a load profile Kn. After a power distribution table **2200** is selected, the system's performance may move up and down across column **2206**. In one embodiment, if the performance of the

system requires an increase in total power, the performance moves up across the column e.g., from system's setting M to system's setting **1**. In one embodiment, entries A**1** . . . Am, B**1** . . . Bm, and C**1** . . . Cm into table **2200** may be provided from the characteristics of the components from which each of the subsystems **1** . . . N is built. For example, a palette of frequencies, and/or core voltages may be used to produce power entries for CPU and GPU.

[**0205**] In one embodiment, the amounts of power in table **2200** may be represented in power units, e.g., watts, milliwatts, and the like, or in arbitrary units. In another embodiment, instead of the amounts of power in power units, table **2200** may include various amounts of current needed to operate the system and the needs of each of the subsystems at different system settings represented in units of current, e.g., amperes, milliamperes, and the like. In alternate embodiments, table **2200** may include various operating frequencies or voltages that correspond to different systems settings.

[**0206**] FIG. **23** is a flowchart of one embodiment of a method to dynamically redistribute power while tracking a load profile of a system when another subsystem is added to the system. The method begins with operation **2301** of adding another subsystem to a plurality of subsystems. In one embodiment, another subsystem, e.g., a second CPU, or a peripheral device, is added to the system which already includes a first CPU and a GPU. Adding another CPU or a peripheral device to the processing system is known to one of ordinary skill in the computer art. Next, in operation **2302** identifying another load profile of the system that includes an added subsystem is performed, as described above with respect to FIGS. **19** and **20**. In one embodiment, the another load profile is identified by determining the utilization factor of the added subsystem, as described above with respect to FIGS. **19** and **20**. The load profile is then calculated using the utilization factors for each of the subsystems including the added subsystem, as described above with respect to FIGS. **19** and **20**. Next, in operation **2303**, the power of the system is redistributed between the subsystems based on the another load profile, as described above with respect to FIGS. **19** and **20**. In one embodiment, adding another subsystem may require more power to be allocated to the system. In such a case, the power of the system may be redistributed by selecting a power distribution table associated with another load profile and with more total allocated power.

[**0207**] FIG. **24** illustrates one embodiment of a system to dynamically redistribute the power while tracking a load profile of a system as described above with respect to FIGS. **18-23**. As shown in FIG. **24** system **2400** includes a subsystem **2401**, e.g., a CPU, a subsystem **2402**, e.g., a GPU that may be coupled with a display device, and one or more subsystems **2409**, e.g., one or more I/O controllers coupled to one or more I/O devices, and a microcontroller **2407** coupled to a bus **2410**. Further, system **2400** includes a volatile RAM **2404**, a non-volatile memory **2406**, e.g., a hard drive, ROM **2403**, and a cache memory **2405** coupled to subsystem **2401** which is coupled to bus **2410**. One or more sensors **2408**, as described above with respect to FIGS. **4**,**5**, **12**, and **17** are coupled to subsystems **2401**, **2402**, **2409**, and to microcontroller **2407**, as shown in FIG. **24**. The sensors may be used to measure or estimate actual power usage by one or more of the subsystems, and the sensors in turn provide the determined power usage values to the microcontroller which may calculate the utilization factors and the corresponding load profile and use the corresponding load profile to select a power distribution table from the plurality of power distribution tables. Components of the system **2400**, including processors, microcontrollers, buses, I/O controllers, I/O devices, memories, sensors are described in detail above with respect to FIGS. **1-17**. In one embodiment, a plurality of power distribution tables corresponding to various load profiles as described above with respect to FIGS. **21**A-**21**C, and **22**, may be generated by subsystem **2401**, and stored in any of memories **2406**, **2404**, and **2405** or within a memory in the microcontroller **2407**. In one embodiment, microcontroller **2407** performs methods described above with respect to FIGS. **19-21** using power distribution tables generated when system **2400** was designed. In another embodiment, subsystem **2401**, rather than microcontroller **2407**, performs methods described above with respect to FIGS. **18-20** and in yet another embodiment, subsystem **2401** and the microcontroller **2407** together perform the methods described above with respect to FIGS. **19-20**.

[**0208**] FIG. **25** is a flowchart of one embodiment of a method to adjust a target temperature of a computer system or of a component in the computer system. The method begins with operation **2501** of receiving a signal associated with a temperature control of a component, e.g., a die. In one embodiment, the component of the computer system is coupled to a cooling system, e.g., a heat sink and the component includes an integrated circuit which is a microprocessor. Generally, the heat sink is an object used to take heat away from another object, such as a microprocessor, to stabilize the temperature of the another object. As such, the heat sink can reduce the temperature of the another object. FIG. **26**A illustrates one embodiment of a system having a component **2602**, e.g., a die, coupled to a heat sink **2601**. Heat sink **2601** takes the heat away from component **2602** through increased thermal mass relative, to mass of component **2602**, and through heat dissipation by conduction, convection, and/or radiation. The heat sink may be made of a thermal conducting material, e.g., a metal, e.g., copper, aluminum, and the like metals. To increase a thermal throughput, a thermal interface material (not shown), e.g., a thermally conductive grease or other material that includes, e.g., colloidal silver, may be placed between the component and the heat sink.

[**0209**] As shown in FIG. **26**A, heat sink **2601** includes a flat surface **2603** to ensure a thermal contact with component **2602** to be cooled. As shown in FIG. **26**A, heat sink includes an array of comb or fin like protrusions **2604** to increase the surface contact with the air that may increase the rate of the heat dissipation. The heat sink may be coupled to a fan (not shown) to increase the rate of airflow over the heat sink **2601** to increase heat dissipation from the heat sink. Component **2602** may be a microprocessor chip, a CPU, a GPU, a microcontroller chip, a memory chip, and/or any other power handling semiconductor device. In one embodiment, component **2602** may be enclosed in a case. In one embodiment, component **2602** may be a microprocessor chip enclosed in a case, wherein the microprocessor includes a logic circuitry (not shown) including one or more monitors (not shown) that continuously monitor a temperature of component **2602**.

[0210] For example, component **2602** may be a microprocessor enclosed in the case, as produced by Intel Corporation, located in Santa Clara, Calif. If the temperature of component **2602** exceeds a component-specific threshold, above which component **2602** may fail to operate, the logic circuitry included in the microprocessor engages throttles that can slow down the frequency of the microprocessor to avoid a failure. The logic circuitry produces a signal associated with the temperature control of component **2602**, which indicates that the temperature of component **2602** reached the component-specific threshold, e.g. a die-specified threshold. In one embodiment, the signal associated with the temperature control of component **2602** is the low-true signal that asserts the PROCHOT# ("PROCHOT_L") pin of an Intel microprocessor. An assertion of the PROCHOT_L pin is an indirect indication that the temperature of heat sink **2601**, is substantially high, such that the case, and component **2602** enclosed in the case, have reached the maximum temperature (the component-specified threshold) causing the assertion of PROCHOT_L pin signal.

[0211] As shown in FIG. **26**A, one or more temperature sensors **2605** monitor the temperature of computer system **2600**. In one embodiment, the one or more sensors **2605** are coupled to heat sink **2601** to monitor the temperature of heat sink **2601** and these sensors are in turn coupled to a thermal controller which may be microcontroller which also receives the PROCHOT # ("PROCHOT_L") signal (or an indicator of the signal). The one or more temperature sensors and the heat sink which is coupled to these sensors and the thermal controller form a thermal control loop which adjusts cooling operations (e.g. turning a fan or several fans on or off) in response to the sensed temperature of the heat sink. The thermal control loop, through the control of the thermal controller, seeks to maintain the measured temperature of the heat sink at or below a target temperature, and the thermal controller adjusts the target temperature up or down in response to assertions (and non-assertions) of the signal which is associated with thermal control of the component such as the PROCHOT_L signal which is on the PROCHOT_L pin). The temperature of heat sink is different from the temperature of component **2602**, and/or the temperature of the case that may enclose component **2602**. The difference between the temperature of heat sink **2601** and component **2602**, and/or the case may be determined by a number of factors that include the nature of heat sink **2601**, the thermal interface material between component **2602** and heat sink **2601**, and a quality of component/heat sink assembly. The one or more temperature sensors **2605** measure an integrated temperature of heat sink to provide a temperature control of heat sink **2601**. In one embodiment, one or more temperature sensors **2605** are placed on a back side **2606** of heat sink **2601**, e.g., a cold plate, which is opposite to surface **2603**, as shown in FIG. **26**A. Positioning sensors **2605** on side **2606** of heat sink **2601** opposite to side **2603** provides measuring an integrated temperature of heat sink **2601** (where the temperature of the heat sink is in effect integrated or arranged over time by the physical mass of the heat sink). That is, substantially all variations of the temperature, e.g., related to component **2602**, and/or other components (not shown) of the computer system are integrated into the measured temperature of heat sink **2601**. Because of a substantially large thermal mass of heat sink **2601**, the integrated temperature of heat sink **2601** changes slowly, such that a temperature control loop of heat sink **2601** does not observe fast temperature changes that are observed by an on-chip thermal sensor. Therefore, the temperature control loop of heat sink **2601** does not require a filter to filter out the fast temperature changes.

[0212] FIG. **26**B illustrates a model of the thermal behavior of a heat sink. As shown in FIG. **26**B, the system includes a heat source **2611**, e.g., a die, coupled to a heat storage **2612**, e.g., a heat sink. Heat storage **2612** may be considered the thermal inertia of the block of metal of the heat sink. Heat storage **2612** functions as a heat capacitor. As shown in FIG. **26**B, heat storage **2612** is coupled to a heat resistor **2613**, e.g., a fin of the heat sink. As shown in FIG. **26**B, heat resistor **2613** is coupled to air **2614**. The temperature measurement may be taken at position **2615** between heat resistor **2613** and heat storage **2612**, as shown in FIG. **26**B to filter out fast variations of the temperature while maintaining the measurement of the integrated temperature. Position **2615** of one or more sensors to measure the temperature of the system **2610** is chosen to maintain the balance between an integration over time and accuracy of the measurement to provide a stable and accurate temperature control loop. Referring back to FIG. **26**A, for example, if one or more sensors **2605** are placed on surface **2603** of heat sink **2601** close to component **2602**, fast variations of the temperature of component **2602** may be sensed by sensors **2605** such that the integrated temperature of the heat sink is not measured. In addition, fast variations of the temperature of component **2602** add noise to the measured temperature affecting the accuracy. If one or more sensors **2605** are placed too far away from component **2602**, e.g., on one of protrusions **2604** at the edge of heat sink **2601**, the ambient temperature, e.g., air temperature, may impact measuring the integrated temperature of heat sink **3202**. In one embodiment, as shown in FIG. **26**A, one or more sensors **2605** are placed on a back side of heat sink **2601**, e.g., on a cold plate. The cold plate is a portion of heat sink **2601** where the heat energy is absorbed and transferred from the heat sink **2601** to e.g., an outside ambient, and a heat removal apparatus, e.g., a fan. Positioning one or more sensors **2605** on the cold plate provides measuring a substantially stable temperature that reflects an amount of energy absorbed by heat sink **2601** while minimizing the impact of temperature variations of component **2602**.

[0213] The temperature control loop of heat sink **2601** controls, in at least certain embodiments, the temperature of heat sink **2601**, such that the temperature does not exceed a target temperature of the heat sink. The temperature control loop of the heat sink is described in further detail below with respect to FIGS. **27**-**29**. Referring back to FIG. **25**, the method continues with operation **2502** which involves adjusting a target temperature of the heat sink (or of the computer system) based on the signal associated with the temperature control of the component, such as the PROCHOT_L signal. In one embodiment, the target temperature of the computer system is determined by the target temperature of the cooling system, e.g., by the temperature of the heat sink. In one embodiment, the target temperature of the cooling system is the target temperature of heat sink **2601**. The target temperature of heat sink **2601** is adjusted, in at least certain embodiments, to operate component **2602** and the case of component **2602** at a highest possible temperature with minimal cooling. The signal associated with temperature control loop of component **2602**, e.g., an

assertion of PROCHOT_L pin, provides the information to the temperature control loop of heat sink **2602** that component **2602** and the case of component **2602** have reached the highest possible temperature without the need of knowing the exact value of such highest possible temperature.

[**0214**] FIG. **27** is a flowchart of one embodiment of a method of operating an adaptive cooling control system of a computer system. Method begins with operation **2701** of operating an off-chip temperature control loop of the computer system. In one embodiment, the off-chip temperature control loop is a heat sink temperature control loop. In one embodiment, operating the heat sink temperature control loop includes measuring a temperature of the heat sink using one or more sensors placed e.g., on the heat sink, and controlling the temperature of the heat sink to stay just below a target temperature of the heat sink, as described above by adjusting the performance of one or more cooling fans and/or other cooling devices. A thermal controller receives temperature measurements from one or more sensors on the heat sink and decides whether to adjust the performance of the cooling devices by comparing the temperature measured on the heat sink to a target temperature for the heat sink. If the measured temperature is less than the target temperature, the thermal controller can decrease the performance of one or more cooling devices (e.g. fans, which generate noise, may be turned off) or can increase the power of the microprocessor or other components, and if the measured temperature is more than the target temperature then the thermal controller can increase the performance of one or more cooling devices, e.g., turn on or increase the fan speed, or can decrease the power of the microprocessor or other components, e.g., by decreasing the operating voltage and/or operating frequency of the microprocessor and/or other components. Next, at operation **2702**, a signal (e.g. the PROCHOT_L signal) associated with a temperature control loop of the component is received, as described above. In one embodiment, the temperature control loop of the component operates outside of and independent of the cooling system temperature control loop. The temperature control loop of the component sets the target temperature of the temperature control loop of the cooling system, such as the target temperature of the heat sink. At operation **2703**, the target temperature of the cooling system, e.g., the target temperature of the heat sink, is adjusted based on the signal. In one embodiment, adjusting the target temperature of the cooling system is performed using a method described below with respect to FIG. **28**.

[**0215**] FIG. **28** is a flowchart of one embodiment of a method to adjust a target temperature of a heat sink based on a component-specific signal. The method begins with operating a heat sink temperature control loop, as described above. At operation **2802**, a signal associated with a temperature control of a component, e.g., a PROCHOT_L pin, is asserted, as described above. Next, at operation **2803** an inquiry is made whether the signal has been asserted for more than a predetermined fraction of time over a predetermined time interval. FIGS. **29**A-**29**C illustrate signals associated with the temperature of the component according to one embodiment of the invention. As shown in FIGS. **29**A-**29**C, each of the signals has a duration T_{signal}. The duration of the signal indicates for how long the signal, e.g., PROCHOT_L pin is asserted. The duration of the signal T_{signal} may be compared with a predetermined fraction of time over a substantially long averaging time interval T_{int}, as shown in FIGS. **29**A-**29**C. The signal may be asserted at any time over the time interval T_{int} and have any duration, as shown in FIGS. **29**A-**29**C. In one embodiment, the ratio T_{signal} to T_{int} may be in the approximate range of 0.001-0.99 depending on the design of the heat sink and the component. As shown in FIG. **29**C, the signal having duration T_{signal} may be asserted a number of times during the time interval Tint. Referring back to FIG. **28**, if the signal is asserted for more than a predetermined fraction of time over a predetermined time interval, a target temperature of the heat sink is decreased. The target temperature may be adjusted by the factor that is a system dependent function of the averaging time, the accuracy of the sensors, and other system dependent features. In one embodiment, the amount of adjustment of the target temperature is determined by the accuracy of temperature sensors. In one embodiment, the target temperature (if the signal is asserted for more than a predetermined fraction of time over a predetermined time interval), of the heat sink may be decreased by about 0.5% to about 30%. In one embodiment, if the signal is asserted for more than e.g., 0.5 seconds to 3 seconds over e.g., 0.5 minutes to 20 minutes, a target temperature of the heat sink is decreased by e.g., 0.5 degree to 3 degree centigrade. Further, method **2800** goes back to operation **2801**. In one embodiment, a predetermined fraction of time for the signal to be asserted may be in the approximate range of 0.5 seconds to 3 seconds and the predetermined time interval over which the predetermined fraction of time is determined, may by in the approximate range of 0.5 minutes to 20 minutes. In one embodiment, the temperature of the heat sink is maintained just below the point at which the signal associated with the temperature control of the component, e.g., a PROCHOT_L pin having a duration, e.g., in the approximate range of 0.5 seconds to 3 seconds is asserted frequently, e.g., not less than 2-5 times over an averaging time interval, e.g., in the approximate range of 0.5 minutes to 20 minutes. In another embodiment, the temperature of the heat sink is maintained just below the point at which the signal associated with the temperature control of the component, e.g., a PROCHOT_L pin, is asserted during a substantially long time (e.g., has a substantially long duration, e.g., in the approximate range of 0.5 seconds to 3 seconds over an averaging time interval, e.g., in the approximate range of 0.5 minutes to 20 minutes. In yet another embodiment, the temperature of the heat sink is maintained just below the point at which the signal associated with the temperature control of the component, e.g., a PROCHOT_L pin having a duration, e.g., in the approximate range of 0.5 seconds to 3 seconds is asserted frequently, e.g., not less than 2-3 times and has a substantially long duration over an averaging time interval, e.g., in the approximate range of 0.5 minutes to 20 minutes. If the signal is not asserted for more than a predetermined fraction of time, e.g., in the approximate range 0.5 seconds to 3 seconds, over the predetermined time interval, e.g., 0.5 minutes to 20 minutes, the target temperature may be optionally increased at operation **2805**. In one embodiment, the target temperature may be increased by about 0.5% to about 30%. In one embodiment, if the signal is asserted for less than e.g., 0.5 seconds to 3 seconds over e.g., 0.5 minutes to 20 minutes, a target temperature of the heat sink is increased by e.g., 0.5 degree to 3 degree.

[**0216**] In another embodiment, if the signal is asserted for less than a predetermined fraction of time, e.g., in the approximate range 0.5 seconds to 3 seconds, over the

predetermined time interval, e.g., 0.5 minutes to 20 minutes, the target temperature is not adjusted. For example, at certain workloads, or if a fan is temporarily obstructed, the component may generate the signal associated with the temperature control of the component e.g. PROCHOT_L no matter what temperature the heat sink is for small bursts, because it can not get the heat off the component quickly enough. In such cases the target temperature may not be adjusted. Further the method continues with operation **2801**. That is, if the signal, e.g., the PROCHOT_L pin, asserts rarely, the control system maintaining the heat sink temperature optionally increases the target temperature to operate the computer system with as little cooling as possible. As such, the temperature control system of the heat sink learns the temperature that corresponds to the minimal amount of cooling, independent of any small variations of the temperature in the component and other components of the system. The temperature control of the computer system dynamically reacts to environmental changes. By dynamically increasing or decreasing a target temperature of the computer system, the temperature control system dynamically adjust key parameters of the computer system, e.g., an acoustics parameters, e.g. a speed of a fan coupled to the heat sink, and/or a temperature of the computer system for a best case operation. That is, the computer system can operate with maximized efficiency at minimal cooling. In one embodiment, operating with minimal cooling with maximized efficiency increases a gradient of the heat across the heat sink. The increased gradient of the heat increases removal of the heat from the heat sink with less air flow. As a result, the cooling system may be operated more efficiently acoustically. For example, if a fan is coupled to the heat sink, increased heat gradient across the heat sink may result in operating the fan with a decreased speed. In one embodiment, the computer systems having the cooling system described with respect to FIGS. **25-29** may be, e.g., a small desktop computer, such as a Mac Mini, or a small laptop such as a small Power Book that are produced by Apple Computer, Inc., located in Cupertino, Calif.

[**0217**] FIG. **30** is a flowchart of another embodiment of a method of using a component-specific signal in a cooling system that includes a heat sink, as described above with respect to FIG. **26**. Method **3000** determines or estimates a difference between the temperature of heat sink **2601** and component **2602** when component **2602** is at a maximum temperature. Method **3000** begins with operation **3001** of increasing a temperature of a heat sink until a signal associated with a temperature control of a component, e.g., a PROCHOT_L pin, is asserted. In one embodiment, the system is configured in such a way that thermal throttle that typically causes the component, e.g., a CPU, to slow down when PROCSHOT_L pin is asserted, is not activated, such that the component, e.g., a CPU, continues to operate at the same frequency as before the signal is asserted. In one embodiment, a maximum temperature of the component at which the signal associated with the temperature control of the component is asserted, is measured. In one embodiment, the temperature of the component may be measured using one or more sensors positioned on the component. Method continues with operation **3002** of measuring the temperature of the heat sink after the signal, e.g., the PROCHOT_L pin, is asserted. In one embodiment, the temperature of the heat sink may be measured using one or more temperature sensors coupled to the heat sink, as described above with respect to FIG. **26**. Next, at operation **3003**, a difference (delta) between a measured temperature of the heat sink and the maximum temperature of the component is computed. The measured temperature of the heat sink may be subtracted from the maximum component temperature. The difference between the temperature of the heat sink and the maximum component temperature provides a baseline for an efficient operational point of a computer system. Method continues with operation **3004** of setting a baseline target temperature of the heat sink based on the difference. In one embodiment, the tolerance parameters for the heat sink/fan assembly may be set based on the computed difference between the temperature of the heat sink and the maximum component temperature. In one embodiment, a compensation for an ambient temperature may be provided, because the thermal resistance of a cooling system may not be linear with ambient temperature. For example, a plurality of measurements at a plurality of temperatures may be performed to compensate for the ambient temperature.

[**0218**] FIG. **31** is a flowchart of one embodiment of a method to operate a cooling system that includes a heat sink. Method begins with operation **3101** of monitoring a temperature of a heat sink using one or more sensors, as described above with respect to FIG. **26**. A component is coupled to the heat sink, as described above with respect to FIG. **26**. At operation **3102**, a signal associated with a temperature control of the component, e.g., a PROCHOT_L pin, is asserted, as described above with respect to FIGS. **25**, **27**, and **28**. Next, at operation **3103** a target temperature of the heat sink is adjusted based on the asserted signal, as described above with respect to FIGS. **25**, **27**, and **28**. Next, at operation **3104** an operation of the component, a cooling unit, e.g., a fan coupled to the component, or both, is adjusted based on a relationship between the monitored temperature of the heat sink and an adjusted target temperature of the heat sink. In one embodiment, adjusting the operation of the component includes changing an operating frequency of the component, an operating voltage of the component, or both, and adjusting the cooling unit is performed by changing a speed of a fan.

[**0219**] FIG. **32** illustrates one embodiment of a computer system **3200** having an adaptive cooling arrangement, as described above with respect to FIGS. **25-31**. As shown in FIG. **32** system **3200** includes a component **3201**, e.g., a CPU, a microprocessor, a GPU, a microcontroller, or any combination thereof. As shown in FIG. **32**, component **3201** is coupled to a cooling system **3210**. As shown in FIG. **32**, cooling system **3210** includes a heat sink **3202** coupled to a fan **3211**, one or more sensors **3203** coupled to heat sink **3202** to measure and monitor temperature of heat sink **3202**, and a power manager **3204**, e.g., a microprocessor, to perform methods described above with respect to FIGS. **25-31**. The power manager **3204** may also be referred to as thermal controller. A position of the one or more sensors **3202** in cooling system **3210** may be chosen to maintain the best compromise between measuring a stable and accurate temperature control loop, as described above with respect to FIGS. **26**A and **26**B. In one embodiment, one or more sensors **3203** are placed on a back side of heat sink **3202**, e.g., on a cold plate. As shown in FIG. **32**, component **3201**, power manager **3204**, e.g., a microcontroller, a subsystem **3205** that includes e.g., one or more I/O controllers coupled to one or more I/O devices, are coupled through a bus **3209**. Bus **3209** may include one or more buses connected to each

other through various bridges, controllers and/or adapters as is well known to one of ordinary skill in the art of computer systems. As shown in FIG. **32**, a volatile RAM **3207**, a non-volatile memory **3208**, e.g., a hard drive, and ROM **3206**, are coupled to power manager **3204**, component **3201** and subsystem **3205** through bus **3209**. In one embodiment, power manager **3204** receives a signal associated with a temperature control of component **3201** and adjusts a target temperature of heat sink **3202** based on the received signal, as described with respect to FIGS. **25-31**. In one embodiment, power manager **3204** operates a temperature control loop of heat sink **3202**. In one embodiment, power manager **3204** increases the temperature of heat sink **3202** to receive the signal associated with the temperature control of component **3201**, measures the temperature of heat sink **3202**, and determines a difference between a measured temperature of heat sink **3202** and a maximum temperature of component **3201**, as described above with respect to FIG. **30**. In one embodiment, power manager **3204** sets a baseline target temperature of heat sink **3202** based on the determined difference. In another embodiment, power manager **3204** sets tolerance parameters and/or validates the tolerance parameters of the cooling system based on the determined difference. The power manager **3204** may also perform one or more of the methods described in connection with FIGS. **1-24**.

[**0220**] FIG. **33** is a flowchart of one embodiment of a method to manage the power of a computer system that leverages intermediate power points. The computer system includes one or more components ("subsystems"). The subsystem may be a microprocessor, a microcontroller, a memory, a CPU, a GPU, or any combination thereof. The method begins with operation **3301** of operating a subsystem at one or more performance points. A performance point may be, e.g., an operational frequency that may define an operational speed of the subsystem, a temperature, or a combination thereof. For example, a processor may be operated at a set of frequencies, e.g., at 600 MHz, at 1 GHz, and at 2 GHz at a specified temperature, e.g., a room temperature. The subsystem is operated at well-known conditions at the performance point. For example, the subsystem may be operated to consume the maximum power at the performance point. In one embodiment, the well-known conditions are those that are substantially close to the thermal design point ("TDP") for a worst-case part of the subsystem. For example, the subsystem, e.g., a processor, may be operated with appropriately chosen software, e.g., a known real world application, or a diagnostic software built for testing the processor, e.g., a power grading software. The method continues with operation **3302** of measuring the actual power consumed by the subsystem at each of the one or more performance points. The power may be measured using a precision power measuring circuitry, e.g., one or more sensors, described above with respect to FIGS. **12**, **24**, and **32**. The power measuring circuitry may be built into the subsystem. In one embodiment, the power may be measured at a performance point, which is substantially close to TDP point for the processor, at a temperature at which the processor is operated. The method continues with operation **3303** of determining an operational power of the subsystem based on the measured power of the subsystem at the performance point. For example, a measured actual power consumed by the subsystem, e.g., a processor, is used as an intermediate operational power allocated to the subsystem at a predetermined frequency. That is, the intermediate operational power points are determined on a "per-subsystem" basis and may include operational power margins for the subsystem.

[**0221**] FIG. **38** illustrates one embodiment of a data processing system that leverages intermediate operational power points for a subsystem in a valuable way. As shown in FIG. **38**, a published specification power **3802** for a subsystem, e.g., a CPU, is 80 W. Such a published specification power is a worst-case power value that is valid for a plurality of subsystems. For example, the published specification power **3802** may be a worst-case power value determined from a statistical power distribution curve **3801** of a vast number of sample processors at a maximum frequency of 3.0 GHz. As shown in FIG. **38**, an intermediate operational power point **3803** determined on a "per-subsystem" basis for the CPU at a frequency of 2.5 GHz is 40 W. The intermediate operational power point **3803** is an actual measured power consumed by the subsystem, e.g., the CPU, at a performance point, e.g., at a frequency 2.5 GHz. For example, at the performance point of 1.0 GHz the intermediate operational power point **3803** for the processor may be 20 W. That is, the intermediate operational power point **3803** has an operational margin that is substantially smaller than published specification power **3802**, which may be considered an established, predetermined value that has a worst-case power margin defined from the statistical power distribution curve. When the data processing system operates the CPU at these intermediate power points the additional power (40 W or 60 W) may be used by the data processing system to operate, for example, another subsystem, e.g., a GPU at various performance points. In other words, at a given intermediate (or other) operational point (such as a given operating frequency at a given temperature), the system or subsystem may normally consume only a portion (e.g. 70%) of an amount of power which has been reserved for it under a conservative, worst-case design; this margin is often, in fact, unnecessary and can be used by the system or subsystem. In effect, the difference between worst-case and actual power can be allocated to other subsystems. As a further example, the power values in the power distribution tables of FIGS. **21A-21C** may include some or all or this difference for each subsystem which is controlled through the use of these power distribution tables. As such, the performance efficiency of the computer system may be substantially increased. In one embodiment, the measured actual maximum power consumed by the subsystem is provided to a power look-up table, as described with respect to FIGS. **6-7**, and **21-22**. As such, the power look up table is built on the fly, on a "per-subsystem" basis. In one embodiment, the power look up tables may be built that include the measured actual maximum power consumed by each of the subsystems at a set of performance points to allocate the power for the subsystem at different computer system settings. In one embodiment, the measured actual maximum power values consumed by the subsystem at a set of performance points are entered into the power look-up table as the power values that are allocated to the subsystem at various settings.

[**0222**] FIG. **34** is a flowchart of one embodiment of a method of providing an intermediate processor power margin for a subsystem. The method begins with operation **3401** of operating a subsystem at well-known conditions at a performance point, as described above. The method contin-

ues with operation **3402** of measuring the power consumed by the subsystem at the performance point, as described above with respect to FIG. **33**. Next, determining an operational power of the subsystem using the measured power is performed at operation **3403**. In one embodiment, the operational power may include operational power margins. Further, at operation **3404** the operational power of the subsystem may be optionally adjusted based on another data associated with the subsystem. The another data associated with the subsystem may be a data provided by a feedback circuitry coupled to the subsystem. The feedback circuitry may provide data associated with the temperature the subsystem. For example, the data associated with the subsystem may be a signal associated with a temperature of a die of the subsystem, e.g., an assertion of PROCHOT_L pin, as described above. In one embodiment, the operational power may be adjusted to add extra power. The extra power may be added to include a design margin for measuring error, measuring accuracy, and/or measuring resolution of the power measuring circuitry. For example, if the power measuring circuitry have a measuring error, e.g., in the approximate range of 1 to 5%, the measured operational power may be adjusted to include the measuring error. The extra power may be added to include a design margin for a temperature and a margin to a future higher power code. The adjusted power may be used to provide entries to power look-up tables of the computer system, e.g., as described above with respect to FIGS. **6**, **7**, **21**, and **22**. The adjusted operational power values may be used to operate the subsystem. Next, at operation **3405** the operational power for the subsystem is provided to a power lookup table of a computer system. In one embodiment, the power look up table of the computer system may be a power look up table as described above with respect to FIGS. **6**, **7**, **21**, and **22**. Next, at operation **3406** determination is made whether to operate the system at a next performance point, e.g., at another frequency, another temperature, or both. Operations **3401-3405** are repeated, if the subsystem is needed to operate at the next performance point. If the subsystem is not needed to operate at the next performance point, method **3400** continues with operation **3407** of using the operational power points to operate the subsystem or to store values in power distribution tables for use in machines to be manufactured.

[**0223**] FIG. **35** is a flowchart of another embodiment of a method of using intermediate operational power points to distribute power in a computer system. Method **3500** begins with operation **3501** of operating one or more subsystems of a computer system at well-known conditions at one or more performance points. In operation **3502** the power consumed by each of the one or more subsystems at each of the one or more performance points is measured. In one embodiment, measuring the power consumed by each of the subsystems is performed in parallel. For example, when the subsystems are operated at the same time, the measuring may be performed in parallel using one or more sensors coupled to the one or more subsystems. In another embodiment, measuring the power consumed by each of the subsystems is performed in series. For example, when one subsystem at a time is operated at the well-known conditions, measuring may be performed in series using one or more sensors coupled to the subsystems. Next, determining operational powers for each of the one or more subsystems is performed in operation **3503**. Next, distributing the power among the subsystems of the computer system is performed based on the operational powers of each of the one or more subsystems in operation **3504**. Accurate knowledge of the intermediate powers at each of the performance points allow for designs that carefully adjust the performance of each of the subsystems in the computer system to allow maximum possible performance under a current set of constraints, e.g., thermal or electrical power constraints, for the computer system. The measuring of intermediate operational powers and the incorporating of the results of these measurements provide a platform specific basis for power distribution and balancing in one or more subsystems of the computer system. In one embodiment, the platform specific intermediate operational powers may be determined as a part of the factory test process and written into the system management controller ("SMC") of the computer system for use in the power balancing algorithm. In another embodiment, the platform specific intermediate operational powers may be determined periodically during a life time of a computer system, and stored in SMC. Next, at operation **3504**, distributing the power among the subsystems of a computer system is performed based on the operational powers allocated to each of the subsystems to balance the power in the computer system. In one embodiment, power look up tables may be built that include the measured actual power consumed by each of the subsystems at maximum-power conditions and at a set of performance points to distribute and balance the power among the subsystems at different computer system settings. In one embodiment, the measured actual power values consumed by each of the subsystems at maximum-power conditions and at a set of performance points are entered into the power look-up table as the power values allocated to the each of the subsystems at various settings.

[**0224**] FIG. **36** is a flowchart of one embodiment of a method of determining intermediate operational powers of one or more subsystems of a computer system. This method begins with operation **3601** which involves operating a subsystem of a computer system at well-known conditions at one or more performance points. In operation **3602** the power consumed by the subsystem at each of the one or more performance points is measured to determine operational powers for the subsystem at each of the one or more performance points. Next, determination is made at operation **3603** whether operational power is to be determined for another subsystem. If the operational power is to be determined for another subsystem, method **3600** returns to performing operations **3601-3602** for another subsystem. If the operational power for another subsystem does not need to be determined, method **3600** continues with operation **3604** of distributing the power in the computer system using the operational powers determined based on the measured power of the one or more subsystem.

[**0225**] FIG. **37** illustrates one embodiment of a system that leverages intermediate operational power margins and distributes the power as described above with respect to FIGS. **33-36**. As shown in FIG. **37** system **3700** includes a subsystem A **3701**, e.g., a CPU, a subsystem B **3702**, e.g., a GPU that may be coupled with a display device, subsystem C **3704**, e.g., a memory, subsystem D **3705**, e.g., a microprocessor, and one or more subsystems N **3703**, e.g., one or more I/O controllers coupled to one or more I/O devices, a power manager **3708**, e.g., a microcontroller, a system management controller ("SMC"), coupled to a interconnect **3706**, e.g., a bus. Subsystem C **3704** may be a volatile RAM,

a non-volatile memory, e.g., a hard drive, and/or a ROM. One or more measuring devices **3707**, e.g., one or more sensors as described above with respect to FIGS. **4**, **5**, **12**, **17**, **24**, and **32** are coupled to subsystems **3701-3705**, and to power manager **3708**, as shown in FIG. **37**. A power look-up table **3709** that may include a power distribution table, as described above with respect to FIGS. **7**, **21**, and **22**, is coupled to power manager **3708**, as shown in FIG. **37**. Components of the system **3700**, including processors, microcontrollers, buses, I/O controllers, I/O devices, memories, sensors are described in detail above with respect to FIGS. **1-17**. In one embodiment, one or more power lookup tables corresponding to various performance settings of the computer system as described above with respect to FIGS. **1-36**, may be generated by subsystem **3701** (or generated by test equipment in the design and/or manufacturing process), and stored in memory **3704**, and/or in a memory located in power manager **3708**. In one embodiment, power manager **3708** performs methods described above with respect to FIGS. **33-36**. In another embodiment, subsystem **3701** performs methods described above with respect to FIGS. **33-36**.

[**0226**] In the foregoing specification, the invention has been described with reference to specific exemplary embodiments thereof. It will be evident that various modifications may be made thereto without departing from the broader spirit and scope of the invention as set forth in the following claims. The specification and drawings are, accordingly, to be regarded in an illustrative sense rather than a restrictive sense.

What is claimed is:

1. A method, comprising:

identifying a load profile of a system that has a plurality of subsystems, and

redistributing a power of the system between the subsystems based on the load profile.

2. The method of claim 1, wherein the load profile is an asymmetric load profile and wherein the power is redistributed by setting new allocated amounts of available power usage among the subsystems.

3. The method of claim 1, wherein the identifying the load profile includes

determining a utilization factor for each of the subsystems.

4. The method of claim 1, wherein the redistributing the power includes

selecting a power weighting arrangement between the subsystems in the system.

5. The method of claim 1, wherein the identifying the load profile includes

sensing the power used by each of the subsystems;

determining for each of the subsystems a utilization factor; and

determining the load profile using the utilization factors of a set of each of the subsystems.

6. The method of claim 1, wherein the utilization factor is a ratio of the power used by the subsystem to the power allocated to the subsystem.

7. The method of claim 1, further including

adding an another subsystem to the plurality of the subsystems;

identifying an another load profile of the system that includes the another subsystem; and

redistributing the power of the system between the subsystems based on the another load profile.

8. The method of claim 7, wherein the identifying the another load profile includes

determining a utilization factor for the added subsystem.

9. A method to redistribute a power in a system having a plurality of subsystems, comprising:

(a) determining a power used by each of the subsystems;

(b) determining for each of the subsystems a utilization factor;

(c) determining a load profile of the system using the utilization factors for a set of each of the subsystems; and

(d) selecting a power distribution table that corresponds to the load profile.

10. The method of claim 9, wherein the determining the load profile includes

subtracting the utilization factor of a first subsystem from the utilization factor of a second subsystem.

11. The method of claim 9, wherein the utilization factor is a ratio of the power used by a subsystem to the power allocated to the subsystem and wherein the determining comprises at least one of sensing the power or estimating the power and wherein the power distribution table is used to set new allocated amounts of available power usage among the subsystems.

12. The method of claim 9, further including

repeating (a) to (c) in a sequence;

determining whether a current load profile is different from the load profile determined previously; and

if the load profile is different,

selecting the power distribution table that corresponds to the current load profile.

13. The method of claim 9, further comprising

generating a plurality of power distribution tables corresponding to different load profiles.

14. The method of claim 9, wherein the plurality of the subsystems include

a central processing unit, and

a graphics processor unit.

15. A system, comprising:

a plurality of subsystems,

one or more sensors coupled to the plurality of the subsystems to sense a power used by at least a set of the subsystems; and

a microprocessor coupled to the plurality of the subsystems and to the one or more sensors, wherein the microprocessor is configured to

identify a load profile of the system, and

to dynamically redistribute a power of the system between the subsystems based on the load profile.

16. The system of claim 15, wherein the load profile is identified using a utilization factor for each of the subsystems.

17. The system of claim 16, wherein the utilization factor is a ratio of the power used by a subsystem to the power allocated to the subsystem and wherein the power is dynamically redistributed by setting new allocated amounts of available power usage among the subsystems.

18. The system of claim 15, wherein the subsystems include a central processing unit, and a graphics processor unit.

19. The system of claim 15, wherein the microprocessor is configured to determine a load profile of the system using the utilization factor for each of the subsystems; and

to select a power distribution table based on the load profile.

20. The system of claim 19, wherein the microprocessor is configured to generate a plurality of power distribution tables corresponding to different load profiles.

21. The system of claim 15, further comprising

a memory coupled to the plurality of the subsystems, wherein the memory stores a plurality of power distribution tables corresponding to different load profiles.

22. The system of claim 15, wherein the system is portable.

23. The system of claim 15, wherein the microprocessor is configured to

add an another subsystem to the plurality of the subsystems, and to

identify another load profile of another system that includes the another subsystem; and to redistribute the power of the another system between the subsystems based on the another load profile.

24. An article of manufacture comprising:

a machine-accessible medium including data that, when accessed by a machine, cause the machine to perform operations comprising,

identifying a load profile of a system that has a plurality of subsystems, and

redistributing a power of the system between the subsystems based on the load profile.

25. The article of manufacture of claim 24, wherein the load profile is an asymmetric load profile and wherein the power is redistributed by setting new allocated amounts of available power usage among the subsystems.

26. The article of manufacture of claim 24, wherein the identifying the load profile includes

determining a utilization factor for each of the subsystems.

27. The article of manufacture of claim 24, wherein the redistributing the power includes

selecting a power weighting arrangement between the subsystems in the system.

28. The article of manufacture of claim 24, wherein the identifying the load profile includes

sensing the power used by each of the subsystems;

determining for each of the subsystems a utilization factor; and

determining the load profile using the utilization factors of a set of each of the subsystems.

29. The article of manufacture of claim 24, wherein the utilization factor is a ratio of the power used by the subsystem to the power allocated to the subsystem.

30. The article of manufacture of claim 24, wherein the machine-accessible medium further includes data that cause the machine to perform operations, comprising,

adding an another subsystem to the plurality of the subsystems;

identifying an another load profile of the system that includes the another subsystem; and

redistributing the power of the system between the subsystems based on the another load profile.

31. The article of manufacture of claim 30, wherein the identifying the another load profile includes determining a utilization factor for the added subsystem.

32. An article of manufacture comprising:

a machine-accessible medium including data that, when accessed by a machine, cause the machine to redistribute a power in a system having a plurality of subsystems that comprises,

(a) determining a power used by each of the subsystems;

(b) determining for each of the subsystems an utilization factor;

(c) determining a load profile of the system using the utilization factors for a set of each of the subsystems; and

(d) selecting a power distribution table that corresponds to the load profile.

33. The article of manufacture of claim 32, wherein the determining the load profile includes

subtracting the utilization factor of a first subsystem from the utilization factor of a second subsystem.

34. The article of manufacture of claim 32, wherein the utilization factor is a ratio of the power used by a subsystem to the power allocated to the subsystem and wherein the determining comprises at least one of sensing the power or estimating the power.

35. The article of manufacture of claim 32, wherein the machine-accessible medium further includes data that cause the machine to perform operations comprising,

repeating (a) to (c) in a sequence;

determining whether a current load profile is different from the load profile determined previously; and

if the load profile is different,

selecting the power distribution table that corresponds to the current load profile.

36. The article of manufacture of claim 32, wherein the machine-accessible medium further includes data that cause the machine to perform operations comprising,

generating a plurality of power distribution tables corresponding to different load profiles.

37. The article of manufacture of claim 32, wherein the plurality of the subsystems include

a central processing unit, and

a graphics processor unit.

38. A system, comprising

means for identifying a load profile of a system that has a plurality of subsystems, and

means for redistributing a power of the system between the subsystems based on the load profile.

39. The system of claim 37, wherein the load profile is an asymmetric load profile.

40. The system of claim 37, wherein the means for identifying the load profile include

means for determining an utilization factor for each of the subsystems.

41. The system of claim 37, wherein the means for redistributing the power include

means for selecting a power weighting arrangement between the subsystems in the system.

42. The system of claim 37, wherein the means for identifying the load profile include

means for sensing the power used by each of the subsystems;

means for determining for each of the subsystems an utilization factor; and

means for determining the load profile using the utilization factor of each of the subsystems.

43. The system of claim 37, wherein the utilization factor is a ratio of the power used by the subsystem to the power allocated to the subsystem.

44. The system of claim 37, further including

means for adding an another subsystem to the plurality of the subsystems;

means for identifying an another load profile of the system that includes an added subsystem; and

means for redistributing the power of the system between the subsystems based on the another load profile.

45. The system of claim 43, wherein the means for identifying the another load profile

includes means for determining a utilization factor for the added subsystem.

46. A system to redistribute a power between a plurality of subsystems, comprising:

means for (a) determining a power used by each of the subsystems;

means for (b) determining for each of the subsystems an utilization factor;

means for (c) determining a load profile of the system using the utilization factor for each of the subsystems; and

means for (d) selecting a power distribution table that corresponds to the load profile.

47. The system of claim 46, wherein the means for determining the load profile include

means for subtracting the utilization factor of a first subsystem from the utilization factor of a second subsystem.

48. The system of claim 46, wherein the utilization factor is a ratio of the power used by a subsystem to the power allocated to the subsystem.

49. The system of claim 46, further including

means for repeating (a) to (c) in a sequence;

means for determining whether a current load profile is different from the load profile determined previously; and

means for selecting the power distribution table that corresponds to the current load profile, if the load profile is different.

50. The system of claim 46, further comprising

means for generating a plurality of power distribution tables corresponding to different load profiles.

51. The system of claim 46, wherein the plurality of the subsystems include

a central processing unit, and

a graphics processor unit.

* * * * *