

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
5 May 2011 (05.05.2011)

PCT

(10) International Publication Number  
**WO 2011/051543 A1**

(51) International Patent Classification:  
*G06F 1/32* (2006.01) *H04W 52/02* (2009.01)  
*G06F 9/445* (2006.01)

(74) Agent: **NOKIA CORPORATION**; IPR Department,  
Virpi Tognetty, Keilalahdentie 4, FI-02150 Espoo (FI).

(21) International Application Number:  
PCT/FI2010/050690

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

(22) International Filing Date:  
7 September 2010 (07.09.2010)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
12/606,842 27 October 2009 (27.10.2009) US

(71) Applicant (for all designated States except US): **NOKIA CORPORATION** [FI/FI]; Keilalahdentie 4, FI-02150 Espoo (FI).

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK,

(72) Inventor; and

(75) Inventor/Applicant (for US only): **VOUTILAINEN, Martti Kalevi** [FI/FI]; Pensaskertuntie 17, FI-02660 Espoo (FI).

[Continued on next page]

(54) Title: NONVOLATILE DEVICE

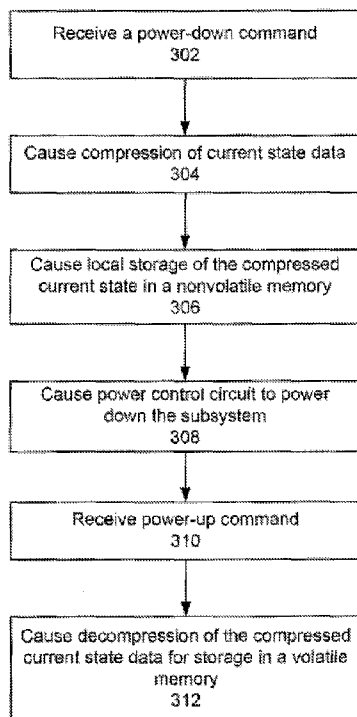


FIG. 3

(57) Abstract: Apparatuses and methods may include receiving a power-down command at a first subsystem comprising a first processor, a first volatile memory, a first nonvolatile memory, a first compressor/decompressor, and a first power control circuit, the first volatile memory being configured to store state data relating to operating conditions of the first subsystem. In response to receipt of the power-down command, the apparatuses and methods may cause the first compressor/decompressor to compress the state data to generate compressed state data and to cause the compressed state data to be stored in the first nonvolatile memory. In response to storage of the compressed state data in the first nonvolatile memory, the apparatuses and methods may cause the first power control circuit to power down the first subsystem.



WO 2011/051543 A1

SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG). **Published:**

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

## NONVOLATILE DEVICE

### 5 BACKGROUND

[1] Consumers desire to have services available on their devices at all times and in all locations. To meet these desires using conventional complementary metal oxide semiconductor (CMOS) technologies, all relevant circuitry of a device always has to be turned on thereby consuming power even when not in use. Always powering  
10 infrequently used circuitry wastes power and detrimentally affects battery performance. Moreover, conventional mobile devices often cannot execute tasks energy-efficiently on general purpose processors. Very simple tasks may be performed by powerful processors, whereas real-time, low-latency tasks may be performed by less powerful processors resulting in undesired delay.

### 15 SUMMARY

[2] The following presents a simplified summary of some example embodiments to provide a basic understanding. This summary is not an extensive overview, and is not intended to identify key or critical elements or to delineate the scope of the claims. The following summary merely presents some concepts and example  
20 embodiments in a simplified form as a prelude to the more detailed description provided below.

[3] Apparatuses and methods can provide improved power consumption by turning off one or more multi-processor subsystems when not in use. The multi-processor subsystems each have a nonvolatile memory storing state data that can be rapidly  
25 retrieved when required to turn back on. The multi-processor subsystems can thus appear to be turned on when, in fact, they are not.

[4] According to at least some aspects, apparatuses and methods can include receiving a power-down command at a first subsystem comprising a first processor, a first volatile memory, a first nonvolatile memory, a first compressor/decompressor, and  
30 a first power control circuit, the first volatile memory being configured to store state data relating to operating conditions of the first subsystem. In response to receipt of the power-down command, the apparatuses and methods may cause the

first compressor/decompressor to compress the state data to generate compressed state data and to cause the compressed state data to be stored in the first nonvolatile memory. In response to storage of the compressed state data in the first nonvolatile memory, the apparatuses and methods may cause the first power control circuit to power down the first subsystem.

#### BRIEF DESCRIPTION OF THE DRAWINGS

[5] A more complete understanding of the present disclosure and the advantages thereof may be acquired by referring to the following description in consideration of the accompanying drawings, in which like reference numbers indicate like features and wherein:

[6] Figure 1 shows an exemplary device having multiple multi-processor subsystems.

[7] Figure 2 illustrates an example flow diagram for decompressing current state data when turning a subsystem back on.

[8] Figure 3 illustrates an example flow diagram a method for powering up and down a subsystem.

#### DETAILED DESCRIPTION

[9] In the following description of the various embodiments, reference is made to the accompanying drawings which form a part hereof, and in which is shown by way of illustration various embodiments in which the invention may be practiced. It is to be understood that other embodiments may be utilized and structural and functional modifications may be made without departing from the scope of the present invention.

[10] Figure 1 shows an example device 100 configured to have its multi-processor subsystems 101A and 101B appear to be turned on when in fact certain subsystems have been turned off. The subsystems 101 can retrieve locally stored state data to rapidly turn back on. The device 100 may be, for example, a mobile phone, a computer, a PDA, a laptop, or other computational device. The device 100 may include a processor 125 and a mass storage device 123 storing computer readable

instructions that, when executed, cause the device 100 to perform the functions described herein. Examples of a mass storage device 123 include a hard disk and a Flash memory, as well as other types of mass storage devices.

5 [11] The device 100 can also include multiple multi-processor subsystems 101. Figure 1, for example, depicts device 100 having two subsystems 101A and 101B, but the device 100 may include any number of subsystems. The device 100 may selectively turn on and off one or more of the multi-processor subsystems 101 at respective times to control power consumption. For example, a mobile phone may turn off a multi-processor subsystem 101 providing a camera when not in use.

10 [12] The multi-processor subsystems 101 may include hardware and/or software optimized to perform a particular computational task based on, for example, latency requirements and processing bandwidth. A subsystem 101 may be a limited-purpose unit that communicates with other subsystems with messages, such as the one described in U.S. Application No. 11/999,314, filed December 4, 2007, titled  
15 “Multi-Processor architecture for a device,” and published as U.S. PreGrant Pub. No. 2009/0141654, the contents of which are incorporated herein by reference in their entirety. Every subsystem 101 may be optimized for one or a limited number of purposes. As such, a subsystem 101 can use very efficient hardware processing instead of software. Hardware processing may use fixed logic with a small amount  
20 of software stored in the nonvolatile memory 104 to give some flexibility. For example, the device 100 may be a mobile phone running a simple operating system (e.g., Symbian) primarily providing a user interface controller. The operating system may process user input (e.g., take a video clip) received by the user interface controller to generate and allocate task requests to optimized subsystems  
25 101.

[13] The device 100 may allocate tasks to the multi-processor subsystems 101 based on performance characteristics required to perform a particular computational task. For example, the device 100 may receive a request to perform a task, may determine which multi-processor subsystem 101 is suited to perform the requested  
30 task, and may cause the identified multi-processor subsystem 101 to perform the task. For example, a subsystem 101A may be optimized for performing

decompression and may receive a task request with instructions to uncompress a predetermined number of data frames into a 1920x1080 pixel picture and send to a display subsystem 101B. As another example, one subsystem 101A may be capable of performing a task requiring real-time, very low latency, and high bandwidth computing, such as for handling of graphics data in game applications. A different subsystem 101B may be capable of performing a task supporting sensor data handling that typically has very long latencies without performance degradation. Example multi-processor subsystems 101 may include supercomputers, video processors, nanotechnology computers, an Audio Codec subsystem, an Audio Codec including a stereo speaker power amplifier with a stereo headphone amplifier.

[14] With embodiments such as those shown in Figure 1, the multi-processor subsystems 101 can be of various types, including: (1) A multipoint bus using time division multiplexing for multiprocessor subsystems for applications that have a small amount of data to be transmitted between processors but has connections between every processor of the multiprocessor unit; (2) switched network (circuit-switched) multiprocessor subsystems using very low-latency point-to-point connections through a cross-bar switch type network; (3) a packet switched network multiprocessor subsystem for applications that tolerate a long latency but require very large bandwidth; and (4) a shared block device for multiple processors. An example implementation of a shared block device is the shared use of a SDRAM or FIFO by several processors running the same operating system. Shared memory allows two or more processes to access common data structures by placing them in a shared memory segment. The shared block device may be used as hardware support or hardware acceleration of OS messaging (e.g., System V Interprocess Communication (IPC)).

[15] The multiprocessor subsystems 101 can be connected to one another through bridge modules 120, in which one side (toward packet network 111) uses a packet network protocol and hardware (e.g., Spacewire or UniPro2) and the other (toward the multi-processor subsystem 101) uses a protocol and hardware specific to the associated multi-processor subsystem 101. An example architecture of the device

100, including bridge modules, such as the one described in U.S. Application No. 11/999,314, filed December 4, 2007, titled "Multi-Processor architecture for a device," and published as U.S. PreGrant Pub. No. 2009/0141654, the contents of which are incorporated herein by reference in their entirety.

- 5 [16] The multi-processor subsystems 101 can connect to the central packet network 111 using an interface such as Spacewire (ECSS-E50-04A) or Mobile Industry Processor Interface (MIPI) Alliance Universal protocol (UniPro2). UniPro2 and Spacewire are only examples of packet network protocols. Other packet network protocols (e.g., Internet) may be used. Many communication methods (e.g., MIPI
- 10 D-PHY and M-PHY used with UniPro2) may have multiple communication modes with different speeds, and the device 100 may instruct a subsystem 101 to use a particular mode. For example, the device 100 may use a squelch system to wake-up a subsystem 101, and specify slow speed or high-speed modes for control and super-speed for high bandwidth transmission.
- 15 [17] A device 100 that is a mobile phone, for instance, can include relatively independent subsystems 101 that communicate via the central packet network 111 using energy-efficient packet-based methods like MIPI UniPro and M-PHY. Central packet network 111 can be connection-less (i.e., data is sent from one end point to another without prior arrangement.) However, other connection-less and
- 20 connection-oriented network protocols can be used.
- [18] While the embodiment shown in Figure 1 includes a single central packet network 111, some embodiments of the invention may utilize a communication network having any number of networks, in which no particular network may be considered to be central since the networks are merely connected to each other. The
- 25 connections may form a chain of networks or a network of networks.
- [19] Each subsystem 101 can include multiple processors 102, a nonvolatile memory 104, a compressor/decompressor 112, a power control circuit 116, and a volatile memory 118. The multiple processors 102 can interact with each other using a bus, packet network, crossbar switch, or other communication path. The volatile
- 30 memory 118 and/or nonvolatile memory 104 can store computer executable

instructions that, when executed by one or more of the multiple processors 102, cause the subsystem 101 to perform the functions described herein. In an example embodiment, the nonvolatile memory 104 may be a spintronic memory composed of graphene transistors. A spintronic memory may also be referred to as a spin torque transfer memory. Companies developing Spin Torque Transfer Random Access Memory (STT-RAM) include EverSpin, Grandis, Hynix, IBM, Samsung, TDK and Toshiba. The nonvolatile memory 104 can also be a Phase-Change memory (PCM) available commercially from Numonyx, a Ferroelectric Random Access Memory (FeRAM), a magnetic flip flop (MFF), or a graphene field effect device.

[20] For example, the nonvolatile memory 104 may be a FeRAM storage capable of storing 128 Megabits and having read/write speeds of 1.6 Gigabytes a second. The nonvolatile memory 104 may also be a MFF where MFF operations are produced by integrating data flip flop (DFF) with magnetic tunnel junctions (MTJ), in addition to circuits that switch the direction of the MTJ's magnetization. During operation, the MFF may be designed to prevent MTJs from affecting MFF clock frequency, as it can operation at the same frequency (e.g., 3.5 GHz) as the DFF. The nonvolatile memory 104 may also be a graphene nonvolatile field effect memory and, in an example, may have a difference of 5 orders of magnitude in resistivity between on and off states. Other nonvolatile memory technologies may also be used.

[21] The power control circuit 116 may power down and up the subsystem 101 in response to commands received from the processor 125 or other subsystems 101. In an example, the power control circuit 116 can include a CMOS inverter and a power switch, where the CMOS inverter which turns on/off a power switch that controls power distribution to the subsystem 101. A CMOS inverter may consume little power (e.g., nanowatt or microwatt). In another embodiment, an Inter-Integrated Circuit (I<sup>2</sup>C) bus can be used. Other types of power control circuits may also be used. The power control circuit 116 may be used to completely turn off the subsystem 101, or there can be several levels of power-saving states, defined, for example, in MIPI D-PHY and M-PHY standards.

- [22] Prior to powering down, a subsystem 101 may store current state data 114 in the nonvolatile memory 104 for quick retrieval and restoration of a previous state in the volatile memory 118 upon being powered back on. Current state data 114 may be stored in the local nonvolatile memory 104 when the subsystem 101 is turned off or when a power saving state would lose important information. Current state data 114 can include operating conditions such as, but not limited to, operation parameters 106, operation code 108, and content 110. The operation parameters 106 may be parameters in which mode communication will be started when next powered-up. Example parameters may be used in hardware-based JPEG compression. The operation code 108 may be executable runtime code. Example operation code includes an operating system (OS), such as, but not limited to, Symbian, Linux, or DOS. Operation code 108 is an example of computer executable instructions. The content 110 may be a Symbian OS.
- [23] The multi-processor subsystems 101 can access their respective nonvolatile memories 104 significantly faster than they can access a mass storage device 123 shared by all of the multi-processor subsystems 101. When powering up, it may take significantly longer for the multi-processor subsystems 101 to load data from the mass storage device 123 as compared with the time required to load data from their respective local nonvolatile memories 104. For instance, if the device 100 is an enterprise personal computer (PC) that has several subsystems 101, but without nonvolatile memories 104, then each would have to load data from the mass storage device 123 when the PC turns on thereby consuming power and time to access the data regardless of how frequently a particular subsystem 101 is used. Providing each subsystem 101 with a local nonvolatile memory 104 advantageously does not require turning on serial links to read data from the mass storage device 123.
- [24] In another example, use of shared transmission channels, such as implementing the packet network 111 as a UniPro2 network, can create latency because due to difficulties in efficiently using available bandwidth when device 100 is composed of both slow and fast subsystems 101 that are dynamically logically connected and disconnected from the device 100 (i.e., powered off/on or even physically disconnected by removing cable). For example, loading a 200MB operating system

using 1Gbps UniPro2 network may only require (200Mbit x 8 = 1600) 1.6 seconds, but when bandwidth is shared with other applications, loading time can significantly increase. When locally storing operating system content in the nonvolatile memory 104 in the same integrated circuit (IC)-package as multiple  
5 processors 102, a 128-bit parallel interface could load 1600 Mbits of operating system content in 62.5ms using a 200 Mbps memory interface. Thus, a subsystem 101 having its own nonvolatile memory 104 can yield significant improvements.

[25] Moreover, a bottleneck could occur in conventional systems when powering up the PC as multiple subsystems 101 might simultaneously attempt to access the mass  
10 storage device 123. Providing each subsystem 101 with its own nonvolatile memory 104 as described herein advantageously decreases the power-up time as the subsystems 101 can simultaneously access their respective local nonvolatile memories 104, instead of competing with one another to access the mass storage device 123 as in conventional systems. A power-up time refers to the amount of  
15 time between when a subsystem 101 is first provided with power to when the subsystem 101 can process data received from or send data via the packet network 111. Further, at start-up, the device 100 can individually power up the multi-processor subsystems 101 as needed while leaving others powered down thus resulting in power savings by not powering unneeded subsystems 101. Moreover,  
20 by having distributed nonvolatile memories 104, a significant amount of data transfer to and from the mass storage device 123 can be avoided. The total amount of storage capacity also can be reduced because each multi-processor subsystem 101 locally stores its data in its nonvolatile memory 104. The reduction can be achieved as the multiple processors 102 of each subsystem 101 are optimized for  
25 tasks they perform and the operating system can be matched to the task and processor hardware.

[26] To turn off a subsystem 101, the device 100 may communicate a power-down command to instruct a particular subsystem 101 to turn off. A subsystem 101 may also communicate power-up and down commands to other subsystems. In an  
30 example embodiment, the device 100 may monitor activity of the subsystems 101 to determine whether to instruct one or more to turn off. The device 100 may issue

a power-down command in response to monitored activity of a subsystem 101 falling below a threshold activity level, the user terminating an application provided by the subsystem 101, expiration of a timer, the user turning off the device 100, or in response to other events. For instance, non-time-critical subsystems 101 may turn off if no tasks are active and no task requests have been received within pre-determined period of time. Prior to turning off, the subsystem 101 can store current state data 114 including operation code 108, content 110, and parameters 106 in its local nonvolatile memory 104, and not in device-wide mass storage device 123. If the local nonvolatile memory 104 is completely full, however, the multi-processor subsystem 101 may store the remaining current state data in the mass storage device 123.

[27] In some embodiments, upon receipt of the power-down command, a subsystem 101 instructs the compressor/decompressor 112 to compress the operation parameters, content, and operation code stored in the volatile memory 118, and the compressor/decompressor 112 causes the compressed operation code, content, and parameters to be stored in the nonvolatile memory 104. Compression can be omitted if a nonvolatile memory 104 has sufficient storage capacity with a fast enough reading time. If the nonvolatile memory 104 has too slow of a reading time, it may be faster to read a small amount of compressed data and then decompress the compressed data with a fast decompression circuit. For example, if the nonvolatile memory 104 has a slow reading time, it may be faster to read a compressed image file and decompress it into a high-resolution picture rather than reading an uncompressed version of the image file.

[28] Locally storing the current state data in a nonvolatile memory 104 may reduce power consumption and the time required to turn back on because the subsystem 101 does not have to retrieve the current state data 114 from the mass storage device 123. Further, a nonvolatile memory 104 can store data even when turned off thereby reducing power consumption of the device 100. For example, reading from mass storage device 123 requires turning on serial links (e.g., UniPro2) between a subsystem 101 and mass storage 123. This can take a long time and requires providing power to both the subsystem 101 and the mass storage 123 during a

transfer that might take seconds. After storage, the subsystem 101 causes the power control circuit 116 to turn off the subsystem 101.

5 [29] Because computational tasks may be assigned to a particular subsystem 101 and performed locally at the assigned subsystem 101, the amount of operation code 108 typically is small enough that local hardware-based compressors/decompressors 112 can compress/decompress the operation code 108 within a limited amount of time (e.g., within one microsecond) giving the user of the device 100 the impression that each of the multi-processor subsystems 101 is on all of the time, when, in fact, they are not. For instance, the local operation code 108 dedicated to hardware and software of a subsystem 101 may include only a few hundred lines of code, but in some cases, operation code for an operating system (e.g., Symbian OS) might be large (e.g., 1GB or more), requiring a large amount of storage in the local nonvolatile memory 104.

10 [30] The compressor/decompressor 112 also may be implemented in software depending on performance requirements. Longer compression/decompression times can be tolerated as long as the computational task to be performed by a subsystem 101 is relatively longer than the time required for compression/decompression to give the user the impression that services provided by the subsystems 101 are available without undue delay. When powered off, the power control circuit 116 may monitor the packet network 111 for a power-up command.

15 [31] At some later time, the device 100 or another subsystem 101 may decide to power back up a subsystem 101 and communicate a power-up command to the subsystem 101 via the packet network 111. For instance, the device 100 may determine that the user has requested a computational task suited to be performed by a powered down subsystem 101. Also, a subsystem 101 may wake up at predetermined time periods when a task is scheduled to be performed. For example, cellular or WLAN modem subsystems may have a predefined timeslot when data is transmitted for a particular user in air interface.

20 [32] When powering back up, the subsystem 101 may retrieve the compressed current state data 114 from the nonvolatile memory 104 and instruct the

30

compressor/decompressor 112 to perform decompression to recover the current state data. The subsystem 101 may then cause the current state data to be stored in the volatile memory 118 to recover its state before it was powered down. For example, the current state data 114 can be uncompressed and loaded into the volatile memory 118 that may be a static random access memory (SRAM). Also, the volatile memory 118 can be omitted and the current state data can be uncompressed and used directly from nonvolatile memory 104 if designed to function as executable memory similar to a RAM.

5  
10  
15  
[33] Figure 2 illustrates an example flow diagram for decompressing current state data 114 when turning a subsystem 101 back on. The depicted embodiment provides ultra-fast decompression of compressed operation code stored in a serial format using Terahertz (THz) grapheme transistors and conversion into parallel data for a 1 gigahertz (GHz) CMOS system. Traditionally, decompression is done using parallel data and parallel processors using wide parallel buses running at 100 MHz - 4 GHz (e.g., in PCs).

20  
25  
30  
[34] The compressor/decompressor 112 may perform computations using serial 1-bit data content and 1-bit code at an ultra high frequency. The compressor/decompressor 112 can directly perform compression or decompression on data (e.g., video) with serial data coming from a nonvolatile memory IC having an ultra high frequency serial port. For example, the ultra high frequency serial port provides data at 5 – 50 Gbps or even at Tbps bit rates or higher. Advantageously, performing computations using serial 1-bit data content and 1-bit code at an ultra high frequency permits use of ultra-fast computing elements, and the serial data is then converted to a parallel form for use by CMOS processors. Because current ultra-fast computing elements may not yet be mature enough for wide-scale mass production similar to silicon CMOS, special computing structures for the compressor/decompressor 112, as well as the other processing units (e.g., processors 102, etc), may be composed of a small number of active elements, such as, but not limited to, graphene/nanotube FET-type transistors, Carbon Nanotubes (CNT), or resonance tunnel diodes (RTD) based logic gates.

- [35] In block 202, the compressor/decompressor 112 may retrieve decompressor operation code from operation code 108 stored in nonvolatile memory 104. In block 204, the compressor/decompressor 112 may retrieve current state data 114 stored in the nonvolatile memory 104 in a compressed format at a rate of 100  
5 Gigabits per second (Gbps). In block 206, the compressor/decompressor 112 may decompress the compressed current state data at a rate of 100 Gbps. In block 208, the compressor/decompressor 112 may perform serial to parallel conversion of the decompressed data at a rate of 100 Gbps to output parallel 1 Gbps of decompressed data. In some systems, CMOS processors and other computing structures are  
10 designed to use parallel data. Conversion of decompressed content from a serial format into parallel format permits use by CMOS processors and other computing structures.
- [36] To further reduce power consumption, the device 100 may limit the amount of information communicated between multi-processor subsystems 101 via the packet  
15 network 111. When a multi-processor subsystem 101 performs a task that changes any of operation parameters, content, and operation code used by at least one other subsystem 101, the multi-processor subsystem 101 may communicate a change message via the packet network 111 to the at least one other subsystem 101. The change message may include only the information describing the change (“change  
20 information”) in a compressed format, and not all of the data affected by the change.
- [37] In an example, a device 100 may be a mobile phone having a video camera subsystem 101A and a display subsystem 101B. The video camera subsystem 101A initially captures image content data and communicates the image content  
25 data to the display subsystem 101B for display. Over time, the video camera subsystem 101A captures new image content data and determines a difference with the earlier image content data. The video camera subsystem 101A then communicates change data (e.g., changed pixels), and not all of the new image content data, to the display subsystem 101B. The display subsystem 101B then  
30 modifies the earlier image content data using the change data to generate updated image content data for display. This method may also be used for other types of

devices that display content, such as, but not limited to, digital television terrestrial/cable/satellite content transmission that communicates pictures at predetermined intervals (e.g., every 20ms) and reference frames less frequently. The changes to a reference frame may not be sent until a next reference frame is sent.

5

[38] Figure 3 illustrates an example flow diagram of a method for powering up and down a subsystem. In block 302, the method may include receiving a power-down command. The device 100 may determine that one of its subsystems 101 is infrequently used and can be powered down. The device 100 may communicate a power-down command to the subsystem 101 via the packet network 111.

10

[39] In block 304, the method may include causing compression of current state data for the subsystems. The subsystem 101 may cause current state data to be provided to the compressor/decompressor 112 from the volatile memory 118 along with a compress instruction to cause the compressor/decompressor 112 to compress the current state data. For example, a current state data could be an operating system (OS), communication parameters specifying what bit rate to use when powered-up next, and a screensaver photograph to display.

15

[40] In block 306 the method may include causing local storage of the compressed current state data in a nonvolatile memory. The compressor/decompressor 112 may communicate the compressed current state data for storage in nonvolatile memory 104.

20

[41] In block 308, the method may include causing a power control circuit to power down the subsystem 101 in response to the compressed current state data being stored in the nonvolatile memory 104.

25

[42] In block 310, the method may include receiving a power-up command. The device 100 may determine that a subsystem 101 is suited to perform a certain computational task, and may communicate a power-up command to that subsystem 101. The power control circuit 116 may cause the subsystem to turn on in response to receiving the power-up command. The power-up command may identify the

type of task to be performed or the subsystem 101 may inform the device 100 of the task after it has powered up.

5 [43] In block 312, the method may include retrieving and decompressing current state data to restore the state of the subsystem in response to receiving the power-up command. The subsystem 101 may cause the compressor/decompressor 112 to retrieve compressed current state data 114 from the nonvolatile memory 104 and decompress the current state data. The compressor/decompressor 112 may then cause the current state data to be stored in the volatile memory 118 to recreate the previous state prior to turning off. The method of Figure 3 may then end.

10 [44] Accordingly, the device 100 can instruct one or more multi-processor subsystems 101 not currently being used to turn off, thereby reducing static power consumption that would otherwise occur when a particular subsystem 101 is turned on even though not currently in use. A nonvolatile memory 104 may locally store current state data 114 allowing a subsystem 101 to quickly restore its previous state to appear as if it is turned on even though powered off.

15 [45] As can be appreciated by one skilled in the art, a computer system with an associated computer-readable medium containing instructions for controlling the computer system can be utilized to implement the exemplary embodiments that are disclosed herein. The computer system may include at least one computer such as a microprocessor, digital signal processor, and associated peripheral electronic circuitry.

20 [46] While the invention has been described with respect to specific examples including presently preferred modes of carrying out the invention, those skilled in the art will appreciate that there are numerous variations and permutations of the above described systems and techniques that fall within the spirit and scope of the invention as set forth in the appended claims.

25

We Claim:

1. An apparatus comprising:

a first subsystem comprising a first processor, a first volatile memory, a first nonvolatile memory, a first compressor/decompressor, and a first power control circuit, the first volatile memory being configured to store state data relating to operating conditions of the first subsystem;

a second processor coupled to the first subsystem, wherein the second processor is configured to generate a power-down command to the first subsystem upon determining that the first subsystem is not needed; and

wherein the first processor is configured to perform:

upon receipt of the power-down command, causing the first compressor/decompressor to compress the state data to generate compressed state data, to cause the compressed state data to be stored in the first nonvolatile memory, and to cause the first power control circuit to power down the first subsystem.

2. The apparatus of claim 1, wherein the first processor is further configured to perform:

upon receiving a power-up command from the second processor, causing the first compressor/decompressor to decompress state data from the first nonvolatile memory and to cause the decompressed state data to be stored in the first volatile memory.

3. The apparatus of claim 2, wherein the first subsystem is configured to perform a computation task using the decompressed state data after storage in the first volatile memory.

4. The apparatus of claim 1, wherein the first subsystem is configured to:

perform a task that changes the state data; and

communicate, to a second subsystem, a change message identifying the change in the state data, wherein the second subsystem comprises a third processor, a second volatile

memory, a second nonvolatile memory, a second compressor/decompressor, and a second power control circuit.

5. The apparatus of claim 1, wherein the first nonvolatile memory is a spintronic memory.

5 6. The apparatus of claim 1, wherein the first subsystem is configured to communicate a second power-down command to a second subsystem.

7. The apparatus of clam 1, wherein the second processor is configured to monitor an activity level of the first subsystem, and wherein the second processor generates the power-down command in response to the activity level falling below a  
10 threshold activity level.

8. A method comprising:

receiving a power-down command at a first subsystem comprising a first processor, a first volatile memory, a first nonvolatile memory, a first compressor/decompressor, and a first power control circuit, the first volatile memory being configured to store state data  
15 relating to operating conditions of the first subsystem;

in response to receipt of the power-down command, causing the first compressor/decompressor to compress the state data to generate compressed state data and to cause the compressed state data to be stored in the first nonvolatile memory; and

in response to storage of the compressed state data in the first nonvolatile memory,  
20 causing the first power control circuit to power down the first subsystem.

9. The method of claim 8, further comprising:

receiving a power-up command; and

in response to the power-up command, causing the first compressor/decompressor to decompress the state data from the nonvolatile memory and causing storage of the  
25 decompressed state data in the first volatile memory.

10. The method of claim 9, further comprising performing, by the first subsystem, a computation task using the decompressed state data after storage in the volatile memory.

11. The method of claim 8, further comprising:

5 performing, by the first subsystem, a task that changes the state data; and

communicating, to a second subsystem, a change message identifying the change in the state data.

12. The method of claim 8, further comprising communicating, by the first subsystem, a second power-down command to a second subsystem.

10 13. The method of claim 8, further comprising monitoring an activity level of the first subsystem, wherein the power-down command is generated in response to the activity level falling below a threshold activity level.

14. A computer-readable medium having computer-executable instructions that, when executed, cause a processor to perform a method comprising:

15 receiving a power-down command at a first subsystem comprising a first processor, a first volatile memory, a first nonvolatile memory, a first compressor/decompressor, and a first power control circuit, the first volatile memory being configured to store state data relating to operating conditions of the first subsystem;

20 in response to receipt of the power-down command, causing the first compressor/decompressor to compress the state data to generate compressed state data, and to cause the compressed state data to be stored in the first nonvolatile memory; and

in response to storage of the compressed state data in the first nonvolatile memory, causing the first power control circuit to power down the first subsystem.

25 15. The computer-readable medium of claim 14, wherein the computer-executable instructions, when executed, cause the processor to perform:

receiving a power-up command; and

in response to the power-up command, causing the first compressor/decompressor to decompress the compressed state data from the nonvolatile memory and causing storage of the decompressed state data in the first volatile memory.

5 16. The computer-readable medium of claim 15, wherein the computer-executable instructions, when executed, cause the processor to perform a computation task using the decompressed state data after storage in the volatile memory.

17. The computer-readable medium of claim 14, wherein the computer-executable instructions, when executed, cause the processor to:

perform, by the first subsystem, a task that changes the state data; and

10 communicate, to a second subsystem, a change message identifying the change in the state data.

18. The computer-readable medium of claim 14, wherein the computer-executable instructions, when executed, cause the processor to communicate, by the first subsystem, a second power-down command to a second subsystem.

15 19. The computer-readable medium of claim 14, wherein the computer-executable instructions, when executed, cause the processor to monitor an activity level of the first subsystem, wherein the power-down command is generated in response to the activity level falling below a threshold activity level.

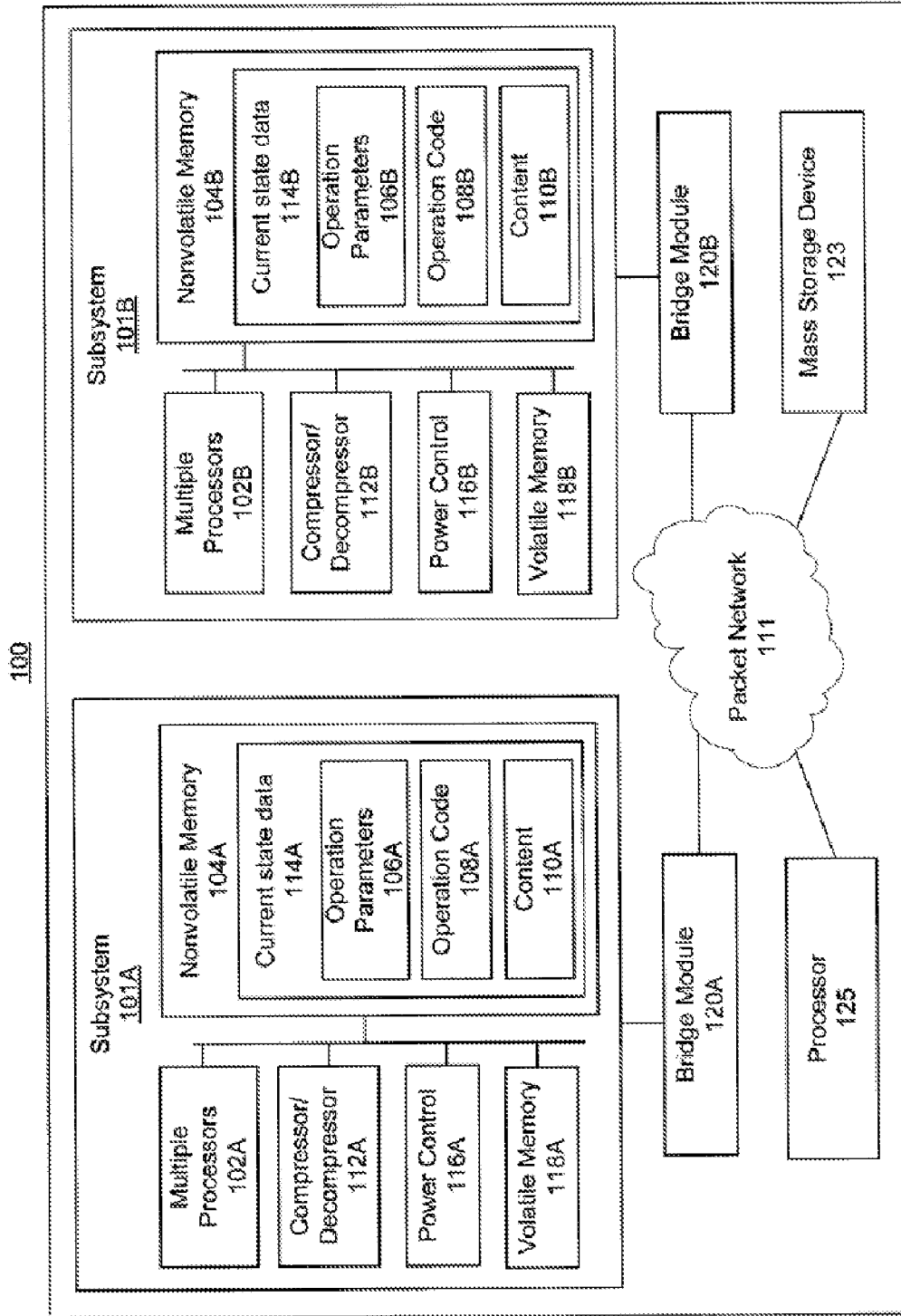


FIG. 1

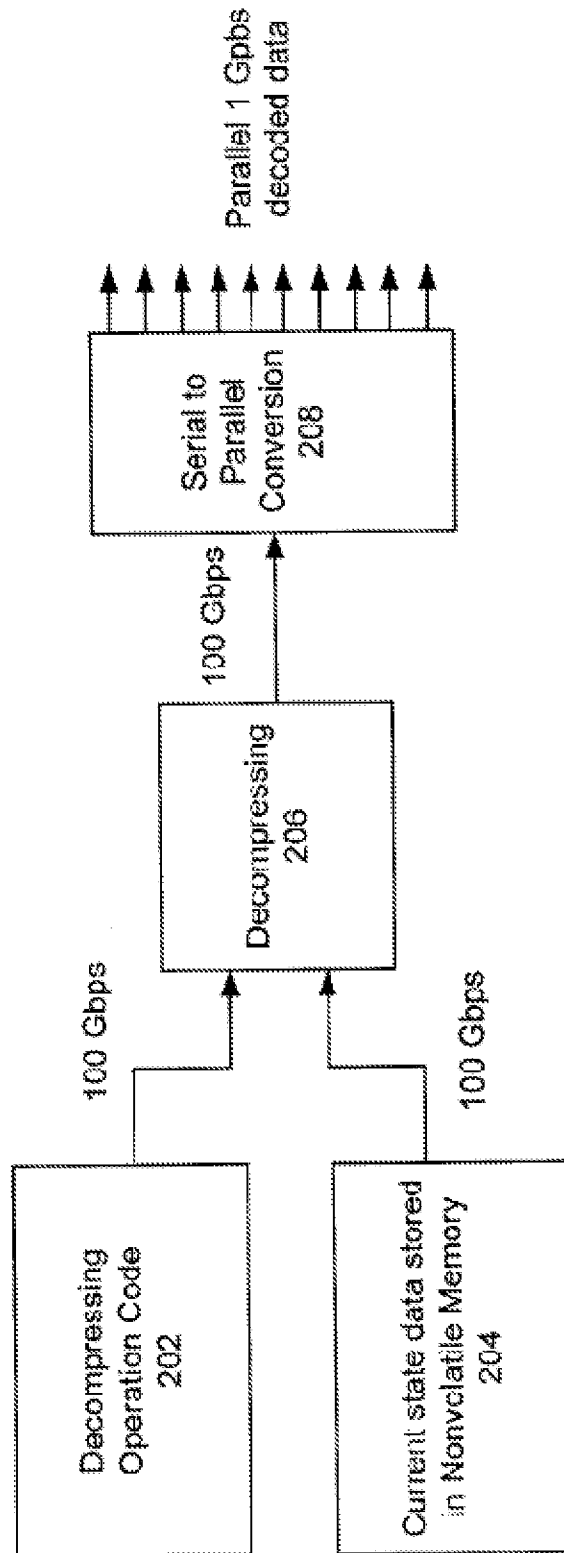


FIG. 2

3/3

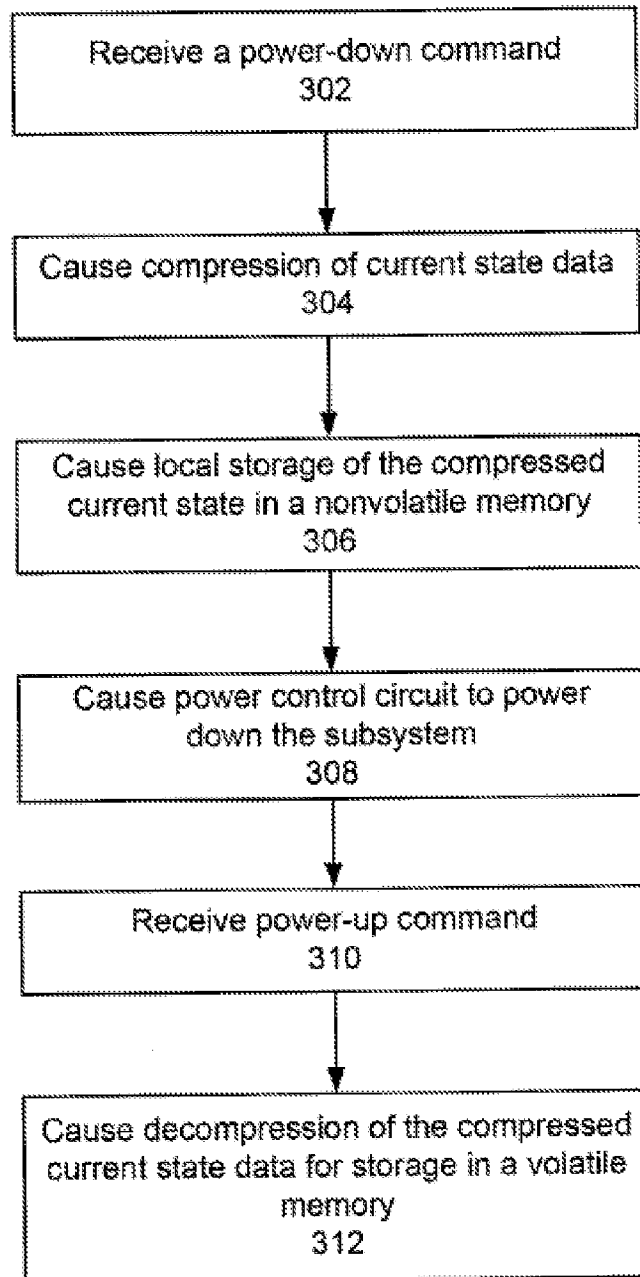


FIG. 3

## INTERNATIONAL SEARCH REPORT

International application No.

PCT/FI2010/050690

A. CLASSIFICATION OF SUBJECT MATTER See extra sheet According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) IPC: G06F, H04L Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched FI, SE, NO, DK Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) EPO-Internal, WPI		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 2009009388 A (SONY ERICSSON MOBILE COMM JP) 15 January 2009 (15.01.2009) abstract [online] EPOQUENET EPODOC & WPI & machine translation into English by Thomson Scientific [online] EPOQUENET TXTJPT	1-19
A	US 2007032228 A1 (VARANDA, M) 08 February 2007 (08.02.2007) the whole document	1-19
A	US 2007291571 A1 (BALASUNDARAM, S P) 20 December 2007 (20.12.2007) the whole document	1-19
P, A	US 2010058007 A1 (YAMAUCHI, S) 04 March 2010 (04.03.2010) paragraph [0017], Claim 1	1-19
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> See patent family annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed		"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family
Date of the actual completion of the international search 07 January 2011 (07.01.2011)		Date of mailing of the international search report 18 January 2011 (18.01.2011)
Name and mailing address of the ISA/FI National Board of Patents and Registration of Finland P.O. Box 1160, FI-00101 HELSINKI, Finland Facsimile No. +358 9 6939 5328		Authorized officer Mari Komulainen Telephone No. +358 9 6939 500

**INTERNATIONAL SEARCH REPORT**  
**Information on patent family members**

International application No.  
PCT/FI2010/050690

Patent document cited in search report	Publication date	Patent family members(s)	Publication date
JP 2009009388 A	15/01/2009	None	
US 2007032228 A1	08/02/2007	US 2009117889 A1	07/05/2009
US 2007291571 A1	20/12/2007	CN 101086680 A GB 2439428 A DE 102007025948 A1	12/12/2007 27/12/2007 03/01/2008
US 2010058007 A1	04/03/2010	JP 4399020B1 B1	13/01/2010

INTERNATIONAL SEARCH REPORT

International application No.  
PCT/FI2010/050690

CLASSIFICATION OF SUBJECT MATTER

Int.Cl.

**G06F 1/32** (2006.01)

**G06F 9/445** (2006.01)

**H04W 52/02** (2009.01)