POWER MANAGEMENT SYSTEM USING BLOCKER MODULES COUPLED TO A BUS

Applicant: Broadcom Corporation, Irvine, CA (US)

Inventors: Mark Fullerton, Austin, TX (US); Lance Flake, Longmont, CO (US); Timothy Chen, Pleasanton, CA (US); Lei Yu, San Diego, CA (US); Anru Wang, San Jose, CA (US); Nirav Pravin Kumar Dagli, Placentia, CA (US); Ronak Patel, Sunnyvale, CA (US)

Assignee: Broadcom Corporation, Irvine, CA (US)

Appl. No.: 14/026,985

Filed: Sep. 13, 2013

Related U.S. Application Data

 Provisional application No. 61/757,947, filed on Jan. 29, 2013.

Publication Classification

Int. Cl.
G06F 1/26 (2006.01)

US. Cl.
CPC ........................................... G06F 1/266 (2013.01)
USPC ............................................ 713/310

ABSTRACT

Systems and methods are provided for efficiently powering down elements and buses of a system without negatively impacting traffic. A power manager receives information from subsystems and determines whether a particular subsystem or bus can be powered down. The power manager sends a message to a subsystem when the subsystem or bus can be safely powered down. Blocker modules are coupled to buses, and the blocker modules respond with an error message if a subsystem attempts to send data over an inactive bus.
FIG. 4

1. Receive request to power down a subsystem or bus (400).
2. Determine if a pending task requires use of the subsystem or bus (402).
   - If yes, leave subsystem or bus powered on (404).
   - If no, configure blocker module to block traffic to subsystem or bus (406).
     - Then power down subsystem or bus (408).
FIG. 5

500 poll a bus to determine pending tasks

502 are there any pending tasks for the bus?

504 initiate request to power down bus and/or subsystem

506 no

508 configure blocker module to recognize a subsystem performing the task as the master of the bus and to block access of other subsystems

510 send a message to subsystem informing it that it is the master of the bus

512 receive a message from the subsystem indicating that the task is complete

504 yes
determine a task with the highest precedence

determine a task with the highest precedence
POWER MANAGEMENT SYSTEM USING BLOCKER MODULES COUPLED TO A BUS

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Patent Application No. 61/757,947, filed on Jan. 29, 2013, which is incorporated by reference in its entirety.

FIELD OF THE INVENTION

[0002] This invention relates to power management and more specifically to power management for modules coupled to a bus.

BACKGROUND

[0003] In many computer systems, buses are used to transfer data between different subsystems. In some computer systems, multiple subsystems can share a single bus. Thus, when one subsystem is using a bus, another subsystem can be blocked from using the bus, which can slow down traffic. Additionally, if a subsystem is powered on but is not being used to perform a task while it is waiting for a bus, the subsystem can waste power.

[0004] Additional problems can arise with shared buses when the state of a subsystem is changed. For example, in systems with multiple power and clock domains, it is often desirable to change the overall state of a particular subsystem, either by changing its power or frequency. Such a change may be a long term change (e.g., a shutdown of a domain) or a temporary change (e.g., a clock change). Changes to the state of a subsystem should ideally be done cleanly (i.e., done while avoiding corruption of bus tasks).

BRIEF DESCRIPTION OF THE DRAWINGS/FIGURES

[0005] The accompanying drawings, which are incorporated in and constitute part of the specification, illustrate embodiments of the disclosure and, together with the general description given above and the detailed descriptions of embodiments given below, serve to explain the principles of the present disclosure. In the drawings:

[0006] FIG. 1 is a block diagram of a system including a power manager, blocker modules, and subsystems in accordance with an embodiment of the present disclosure.

[0007] FIG. 2 is a block diagram of a system including a hub module and multiple subsystems in accordance with an embodiment of the present disclosure.

[0008] FIG. 3 is a block diagram of an exemplary implementation of a system according to an embodiment of the present disclosure.

[0009] FIG. 4 is a flowchart of a method for powering down a subsystem in accordance with an embodiment of the present disclosure.

[0010] FIG. 5 is a flowchart of a method for managing a bus in accordance with an embodiment of the present disclosure.

[0011] FIG. 6 is a block diagram illustrating an example computer system that can be used to implement embodiments of the present disclosure.

[0012] Features and advantages of the present disclosure will become more apparent from the detailed description set forth below when taken in conjunction with the drawings, in which like reference characters identify corresponding elements throughout. In the drawings, like reference numbers generally indicate identical, functionally similar, and/or structurally similar elements. The drawing in which an element first appears is indicated by the leftmost digit(s) in the corresponding reference number.

DETAILED DESCRIPTION

[0013] In the following description, numerous specific details are set forth to provide a thorough understanding of the disclosure. However, it will be apparent to those skilled in the art that the disclosure, including structures, systems, and methods, may be practiced without these specific details. The description and representation herein are the common means used by those experienced or skilled in the art to most effectively convey the substance of their work to others skilled in the art. In other instances, well-known methods, procedures, components, and circuitry have not been described in detail to avoid unnecessarily obscuring aspects of the disclosure.

[0014] References in the specification to “one embodiment,” “an embodiment,” “an example embodiment,” etc., indicate that the embodiment described may include a particular feature, structure, or characteristic, but every embodiment may not necessarily include the particular feature, structure, or characteristic. Moreover, such phrases are not necessarily referring to the same embodiment. Further, when a particular feature, structure, or characteristic is described in connection with an embodiment, it is submitted that it is within the knowledge of one skilled in the art to affect such feature, structure, or characteristic in connection with other embodiments whether or not explicitly described.

[0015] For purposes of this discussion, the term “module” shall be understood to include one of software, firmware, or hardware (such as circuits, microchips, processors, or devices, or any combination thereof), or any combination thereof. In addition, it will be understood that each module can include one, or more than one, component within an actual device, and each component that forms a part of the described module can function either cooperatively or independently of any other component forming a part of the module. Conversely, multiple modules described herein can represent a single component within an actual device. Further, components within a module can be in a single device or distributed among multiple devices in a wired or wireless manner.

1. OVERVIEW

[0016] Embodiments of the present disclosure provide systems and methods for monitoring bus traffic and efficiently shutting down a bus while minimizing negative impacts on bus traffic. A power manager monitors the state of subsystems and of outstanding tasks to or from a subsystem over a bus. The power manager blocks the start of new tasks over the bus and detects when a task requires use of the bus.

[0017] The power manager receives information from subsystems regarding requests to use a bus and/or a subsystem. Based on this information, the power manager determines whether a particular subsystem or bus should be powered down. The power manager sends a message instructing the subsystem or bus to be powered down. In an embodiment, this message can instruct blocker modules to block traffic over the bus. The blocker modules respond with an error message if a subsystem attempts to send data over an inactive bus. In an embodiment, the power manager can also configure the
blocker modules to respond with an error message when more than one subsystem tries to send data over a shared bus.

2. POWER MANAGER

[0018] FIG. 1 is a block diagram of a system including a power manager 102, blocker modules 104, and subsystems 106 in accordance with an embodiment of the present disclosure. In FIG. 1, power manager 102 is coupled to subsystem 106a over bus 108a, power manager 102 is coupled to subsystem 106b over bus 108b, and power manager 102 is coupled to subsystem 106c over bus 108c. Buses 108 can be unidirectional buses or bidirectional buses.

[0019] Power manager 102 monitors the state of subsystems 106 and buses 108 and receives information from subsystems 106 regarding upcoming tasks. In an embodiment, subsystems 106 can send a request to power manager 102 before powering up or powering down. For example, subsystem 106a can send a request to power manager 102 over bus 108a. In an embodiment, power manager 102 can ensure that subsystem 106a will not be needed to perform an upcoming task before granting the request by subsystem 106a to be powered down. If power manager 102 determines that the request should be granted, power manager 102 can send a message (e.g., a control signal) to subsystem 106a over bus 108a instructing subsystem 106a to power down.

[0020] Power manager 102 can also determine when subsystems 106 should be powered up. For example, at any given time, one or more of subsystems 106 can be powered down. If, for example, subsystem 106b requests a task to be performed that requires subsystem 106a to be powered up, subsystem 106a can send a request to power manager 102 over bus 108b for subsystem 106a to be powered up. Power manager 102 can determine whether subsystem 106a should be powered up (for example, power manager 102 may determine that the request should be denied or delayed to conserve power). If power manager 102 determines that the request should be granted, power manager 102 can send a message to subsystem 106a over bus 108a instructing subsystem 106a to power up.

[0021] Alternatively, power manager 102 can initiate powering up the subsystem without first receiving a request to power up the subsystem. For example, power manager 102 can analyze upcoming tasks and can determine that subsystem 106a needs to be powered up to perform an upcoming task. Power manager 102 can then send a message to subsystem 106a instructing subsystem 106a to power up.

[0022] As would be understood by one of ordinary skill in the art, power manager 102 can control the power supply to subsystems 106 using a variety of methods. For example, in an embodiment, one or more of subsystems 106 includes a switching regulator configured to supply power to components of subsystems 106. For example, in an embodiment, power manager 102 can send a message to the switching regulator of subsystem 106a instructing the switching regulator of subsystem 106a to supply power or cut off power to one or more components of subsystems 106a.

[0023] Alternatively, in an embodiment, power manager 102 includes a switching regulator (or is coupled to an external switching regulator), and power manager can supply power or cut off power to each of subsystems 106 (or components of subsystems 106) using this switching regulator. For example, in an embodiment, power manager 102 is coupled to a switching regulator. If for example, power manager 102 wants to cut off power from subsystem 106a, power manager 102 can send a message to the switching regulator instructing the switching regulator to cut off power from subsystem 106a. If power manager 102 determines that subsystem 106a should be powered up again, power manager 102 can send a message to the switching regulator instructing the switching regulator to supply power to subsystem 106a.

3. BLOCKER MODULES

[0024] At any given time, power manager 102 has information regarding upcoming tasks and can determine which subsystems 106 are powered up and which subsystems 106 are powered down. In an embodiment, power manager 102 can use this information to manage buses 108 by configuring blocker modules 104. Power manager 102 can configure blocker modules 104 to block traffic over a bus to a subsystem that has been powered down or to block traffic over a bus when the bus is currently in use. In an embodiment, power manager 102 has access to the functions of blocker modules 104 via a separate control bus (not shown).

[0025] Blocker module 104 can be implemented using hardware, software, or a combination of hardware and software. For example, in an embodiment, blocker modules 104 can be implemented using logic circuitry. In an embodiment, power manager 102 can send a message to blocker modules 104 instructing blocker modules 104 to transition to one of a plurality of states. For example, in an embodiment, blocker modules 104 in a pass state can be configured to allow traffic to pass through buses 108, and blocker modules in a block state can be configured to block traffic from passing through buses 108.

[0026] Alternatively, in an embodiment, power manager 102 can send a message to blocker modules 104 instructing blocker modules 104 which system component is the master of a bus. Based on this information, blocker modules 104 can allow traffic on the bus from the master and can block traffic from other system components. For example, in an embodiment, power manager 102 can send a message to blocker module 104a instructing blocker module 104a that subsystem 106b is the master of bus 108a. Based on this information, blocker module 104c can allow subsystem 106c to send data over bus 108a and block traffic from subsystem 106a: over bus 108a. In an embodiment, blocker module 104a can send an error message to subsystem 106a when subsystem 106a attempts to send data over bus 108a.

[0027] In FIG. 1, blocker modules 104 are positioned at the input of buses 108. However, it should be understood that, in an embodiment, blocker modules 104 can also be positioned at the output of buses 108. Additionally, in an embodiment, multiple blocker modules can be positioned over the length of a single bus. For example, in an embodiment, bus 108a can be coupled to blocker module 104a and an additional blocker module.

3.1 Blocking Traffic to Powered Down Subsystems

[0028] In an embodiment, power manager 102 can instruct blocker modules 104 to block traffic over a bus and to send an error message when a subsystem attempts to send a message to a powered down subsystem. For example, in an embodiment, when subsystem 106a powers down, power manager 102 can send a message to blocker module 104a that instructs blocker module 104a to block traffic over bus 108a and to send an error message to a subsystem (e.g., subsystem 106a).
when subsystem 106b tries to send data to subsystem 106a over bus 108a. For example, this message can instruct blocker module 104a to toggle from the pass state to the block state. By configuring blocker modules 104 to respond with an error message when a subsystem is powered down, power manager 102 can also safely instruct buses 108 to be powered down when they are no longer needed to perform a task. For example, when subsystem 106a powers down, then bus 108a is no longer needed to send data. Thus, power manager 102 can send a message instructing bus 108a to be powered down. Power manager 102 can maintain a power supply to blocker 104a while bus 108a is powered down so that blocker 104a can respond with an error message when a subsystem (e.g., subsystem 106b) attempts to send data over bus 108a.

In an embodiment, power manager 102 can poll a bus to ensure that no current tasks require use of the bus before powering the bus down. For example, power manager 102 can poll bus 108a to determine if there are upcoming tasks that will require the use of bus 108a before powering bus 108a down. Alternatively, the power manager 102 can poll the subsystems 106 to determine if they have upcoming tasks that will require bus 108a.

In an embodiment, power manager 102 can instruct blocker modules 104 to stop blocking traffic over a bus and to stop sending an error message when a subsystem is powered up. For example, in an embodiment, when subsystem 106a powers up, power manager 102 can send a message to blocker module 104a that instructs blocker module 104a to stop blocking traffic over bus 108a and to stop sending an error message to a subsystem (e.g., subsystem 106b) when subsystem 106b tries to send data to subsystem 106a over bus 108a. For example, this message can instruct blocker module 104a to toggle from the block to the pass state.

In an embodiment, blocker modules 104 can detect when buses 108 and/or subsystems 106 are powered up or powered down and can automatically adjust their states in response to detecting a change in bus power. For example, power manager 102 can send a message to subsystem 106a instructing subsystem 106a to be powered down. In an embodiment, blocker module 104a can detect that subsystem 106a has been powered down and can automatically toggle to the block state and initiate powering down bus 108a. In an embodiment, when power manager 102 sends a message to subsystem 106a instructing subsystem 106a to be powered up, blocker module 104a can detect that subsystem 106a has been powered up and can automatically toggle to the pass state and initiate powering up bus 108a.

3.2 Dependent Subsystems

FIG. 2 is a block diagram of a system including a hub module 204 and multiple subsystems 206 in accordance with an embodiment of the present disclosure. In FIG. 2, power manager 102 is coupled to hub module 204 over unidirectional buses 208a and 210a. Blocker module 202a can be configured to block downstream traffic from power manager 102 to hub 204 over bus 208a (e.g., when hub module 204 and bus 208a are powered down).

Hub module 204 is coupled to subsystems 206a, 206d, and 206e. Blocker module 205 can be configured to block downstream traffic from hub module 204 over bus 208a (e.g., when subsystem 206a and bus 208a are powered down). Blocker module 202b can be configured to block downstream traffic from hub module 204 over bus 208c (e.g., when subsystem 206d and bus 208c are powered down). Blocker module 202c can be configured to block downstream traffic from hub module 204 over bus 208c (e.g., when subsystem 206e and bus 208c are powered down).

Subsystem 206b is coupled to subsystems 206a. Blocker module 202a can be configured to block downstream traffic from subsystem 206a over bus 208a (e.g., when subsystem 206b and bus 208c are powered down). Subsystem 206c is coupled to subsystem 206b. Blocker module 202c can be configured to block downstream traffic from subsystem 206b over bus 208c (e.g., when subsystem 206c and bus 208c are powered down). Subsystem 206d is coupled to subsystem 206c. Blocker module 202d can be configured to block downstream traffic from subsystem 206d over bus 208c (e.g., when subsystem 206d and bus 208c are powered down).

In an embodiment, subsystem 206d is not powered down when a dependent subsystem is still powered up. For example, in an embodiment, subsystem 206d is not powered down unless subsystem 206b is powered down because subsystem 206d uses subsystem 206b to perform tasks and/or to send data to other subsystems. Additionally, in an embodiment, subsystem 206a is not powered down unless subsystem 206b is powered down, and subsystem 206d is not powered down unless subsystem 206b is powered down. Further, in an embodiment, hub module 204 is not powered down unless subsystems 206a, 206d, and 206c are all powered down.

In an embodiment, power manager 102 receives information regarding which subsystems are powered up and which subsystems are powered down and confirms that no dependent subsystems require a subsystem to be powered up before authorizing a subsystem to be powered down. For example, in an embodiment, when subsystem 206b sends a request to power down to power manager 102, power manager 102 confirms that subsystem 206c is powered down before authorizing subsystem 206b to be powered down. Additionally, in an embodiment, when power manager determines that subsystem 206c should be powered up, power manager also sends a message instructing hub module 204, subsystem 206a, and subsystem 206b to be powered up if any of hub module 204, subsystem 206a, and subsystem 206b are currently powered down.

In an embodiment, it is not necessary to include a blocker module blocking upstream traffic from subsystem 206b to subsystem 206d over bus 210a because, if subsystem 206c is powered down, no powered up modules are coupled to
For similar reasons, it is not necessary to include a blocker module blocking upstream traffic from hub module 204 to power manager 102 over bus 210a because, if hub module 204 and bus 210a are powered down, no powered up modules are coupled to power manager 102.

By creating a tiered system of subsystems and buses (as illustrated, for example, by FIG. 2), embodiments of the present disclosure enable subsystems and buses to be powered down without negatively impacting performance, and embodiments of the present disclosure ensure that subsystems and buses are powered up when other subsystems require use of these subsystems and buses.

3.3 Blocking Traffic when a Bus is in Use

At any given time, more than one subsystem may attempt to send data over a shared bus. Systems and methods according to embodiments of the present disclosure enable power manager 102 to configure blocker modules 202 to block traffic over a bus when the bus is already in use. For example, subsystem 206d may want to send data to subsystem 206a over bus 208b, and hub module 204 may also want to send data to subsystem 206a over bus 208b. In an embodiment, power manager 102 can configure blocker 202b to block bus 208b while bus 208b is already in use so that both subsystem 206d and hub module 204 do not send data over bus 208b at the same time.

As previously discussed, power manager 102 receives information from subsystems (e.g., hub module 204 and subsystems 206) regarding upcoming operations over buses (e.g., buses 208). For example, subsystems can send information to power manager 102 on a regular basis regarding upcoming tasks. In an embodiment, power manager 102 can also poll buses to determine upcoming bus tasks. For example, power manager 102 can poll bus 208b to determine if there are outstanding operations. Based on this poll of bus 208b, power manager can determine that both subsystem 206d and hub module 204 want to send information over bus 208b.

Once power manager 102 determines that multiple tasks are being attempted over the same bus, power manager 102 can block new tasks. For example, power manager 102 can determine which tasks are pending for a given bus and can determine which of the pending tasks takes precedence. In an embodiment, power manager 102 can determine that tasks should be processed in the order in which they were received. In an embodiment, some tasks may take precedence over other tasks and can be processed earlier even if they were received later (e.g., some tasks may be assigned a higher priority than other tasks, and power manager 102 can determine a priority associated with a task). Power manager 102 can also take into consideration the order in which tasks are received when determining which task to process next. For example, in an embodiment, a task with a low priority might be processed before a task with a higher priority if the lower priority task has been pending for a long time (e.g., if the wait time of the task exceeds a predetermined amount of time). As would be understood by one of ordinary skill in the art, multiple techniques for determining an order for processing tasks can be used in accordance with embodiments of the present disclosure.

Once power manager 102 determines which task should take precedence and thus which task should have access to the bus, power manager 102 sends a message to configure a blocker module to block other subsystems from using the bus. For example, if both subsystem 206d and hub module 204 want to use bus 208b to perform tasks, and if power manager 102 determines that the task hub module 204 takes precedence, power manager 102 can send a message to configure blocker module 202b to block subsystem 206d from sending data over bus 208b until the task initiated by hub module 204 has finished using bus 208b. For example, power manager 102 can send a message to hub module 204 over bus 208a instructing hub module 204 to configure blocker module 202b to block other tasks over bus 208b.

Blockers can be configured to block other tasks from using an active bus in a variety of ways. For example, in an embodiment, blocker 202b can be configured to recognize hub module 204 as the current master of bus 208b and can block other subsystems from using bus 208b. Blocker 202b can be configured to send an error message to other subsystems (e.g., to subsystem 206d) if other subsystems that are not designated as the current master of bus 208b attempt to use bus 208b to send data.

In an embodiment, once a subsystem finishes a task, it informs power manager 102 that the task is completed. Power manager 102 can then poll the bus again to determine if any other subsystems should be allowed to use the bus. For example, when hub module 204 finishes using bus 208b, hub module 204 can send a message to power manager 102. Power manager 102 can then poll bus 208b to determine which tasks are still pending for bus 208b. For example, power manager 102 may determine that subsystem 206d wants to use bus 208b to perform a task. Power manager 102 can then send a message to blocker 202b to instruct blocker 202b to recognize subsystem 206d as the master of bus 208b and to block other subsystems from using bus 208b.

In an embodiment, power manager 102 can designate subsystems as masters of a bus in turn instead of polling a bus after each task has completed. For example, in an embodiment, power manager 102 can poll bus 208b and can determine that both hub module 204 and subsystem 206d want to send data over bus 208b. Power manager 102 can designate, in turn, hub module 204 and subsystem 206d as masters of bus 208b before polling bus 208b again.

In an embodiment, power manager 102 can instruct subsystems to be temporarily shut down to conserve power while another subsystem is using a shared bus. For example, in an embodiment, power manager 102 can instruct subsystem 206d (and buses 208e and 210e) to power down while subsystem 206d is waiting for hub module 204 to finish using bus 208b. When hub module 204 finishes using bus 208b, hub module 204 can send a message informing power manager 102 that its task is complete, and power manager 102 can then send a message to subsystem 206d to instruct subsystem 206d (and buses 208e and 210e) to power up again.

In an embodiment, power manager 102 determines whether it would be power efficient to power down a subsystem while waiting for a task. For example, in some cases, powering down a subsystem temporarily and powering the subsystem back up again might waste more power than the power that would be conserved by powering down the subsystem. Thus, in an embodiment, power manager 102 determines whether power would be saved by temporarily powering down subsystem 206d while subsystem 206d waits for bus 208e before instructing subsystem 206d to power down.

Thus, according to embodiments of the present disclosure, blocker modules 202 can be configured to block traffic over buses 208 when a subsystem is powered down and/or when multiple subsystems attempt to use a shared bus.
to perform a task. In an embodiment, blocker modules 202 can also be used to prevent data from being sent over a bus when software bugs exist. For example, if a subsystem erroneously attempts to send data over a bus to a subsystem that has been powered down (or over a bus that is currently in use), blocker modules 208 can respond with an error message instead of allowing the bug to initiate sending data over the bus. Additionally, because subsystems send messages to power manager 102 when tasks are completed, embodiments of the present disclosure can also prevent errors that might otherwise occur when a subsystem mistakenly determines that a bus is no longer in use and attempts to send data over a bus while another subsystem is still performing a task.

4. EXEMPLARY IMPLEMENTATION

[0055] After blocker module 302b is configured to recognize hub module 308 as the current master of bus 330b, blocker module 302b will respond with an error message if another subsystem attempts to send data over bus 330b. When hub module 308 is finished performing a task using bus 330b, hub module 308 can send a message to power manager 102 informing power manager 102 that the task initiated by hub module 308 has been completed. Power manager 102 can then poll bus 330b again to determine if another subsystem wants to use bus 330b. For example, in an embodiment, power manager 102 can determine that modem module 326 still wants to use bus 330b. Power manager 102 can then instruct subsystem 304g to power up (e.g., by sending a message to LPM 306g) and can send a message to LPM 306a to instruct LPM 306a to configure blocker module 302b to recognize modem module 326 as the current master of bus 330b.

[0056] Additionally, in an embodiment, the components of the systems of FIG. 1, the components of the system of FIG. 2 and/or the components of the system of FIG. 3 can be implemented on a single integrated circuit (IC). In another embodiment, some components of the systems of FIGS. 1, 2, and/or 3 are implemented using multiple ICs. For example, in an embodiment, power manager 102 and the subsystems of FIGS. 1-3 are implemented on different ICs. Additionally, it should be understood that the components of the systems of FIGS. 1, 2, and/or 3 can be implemented using hardware, software, or a combination of hardware and software in accordance with embodiments of the present disclosure.

5. METHODS

[0057] FIG. 4 is a flowchart of a method for powering down a subsystem in accordance with an embodiment of the present disclosure. The method of FIG. 4 will be explained with reference to FIG. 3. In step 400, a request to power down a subsystem or bus is received. For example, when modem module 326 finishes a task, modem module 326 can send a request to power manager 102 to be powered down. In step 402, a determination is made regarding whether a pending task requires use of the subsystem or bus. For example, power manager 102 can determine whether any other subsystems require use of modem module 326. In an embodiment, power manager 102 can poll bus 330f to determine if any tasks are pending for bus 330f that will require data to be sent to subsystem 304g, which includes modem module 326.

[0058] If a pending task requires the use of the subsystem or bus, the method proceeds to step 404, and the subsystem or bus remains powered on. For example, if power manager determines that a task will require data to be sent to modem module 326, power manager determines not to power down subsystem 304g. If a pending task does not require the use of the subsystem or bus, the method proceeds to step 406, and a blocker module is configured to block traffic to the subsystem or bus. For example, power manager 102 can send a message to LPM 306a to instruct blocker module 302a to block downstream traffic to subsystem 304g. Blocker module 302a responds with an error message if a subsystem attempts to send data to subsystem 304g and/or modem 326. In step 408, the subsystem or bus is powered down. For example, power manager 102 can send a message to LPM 306g to instruct subsystem 304g to power down, including modem 326.

[0059] In embodiment, at this time, power manager 102 can also optionally determine to power down the buses coupled to subsystem 304g (e.g., bus 330h since no data will be sent to subsystem 304g or from subsystem 304g while subsystem
304g is powered down. When power manager 102 determines that a task will require the use of subsystem 304g, power manager 102 can instruct subsystem 304g to power up by sending a message to LPM 306f. For example, dual core processor 316 may require use of modem module 326. Power manager 102 then determines (e.g., based on its knowledge of upcoming tasks or in response to a request by dual core processor 316) to power up subsystem 304g and sends a message to LPM 306f. In an embodiment, power manager 102 can then send a message to LPM 306a to instruct blocker module 302c to stop blocking traffic to subsystem 304g. In other words, blocker module 302c will transition from a block state to a pass state for its corresponding bus 330c.

[0060] FIG. 5 is a flowchart of a method for managing a bus in accordance with an embodiment of the present disclosure. The method of FIG. 5 will be explained with reference to FIG. 3. In step 500, a bus is polled to determine pending tasks that require use of the bus. For example, in an embodiment, power manager 102 can poll bus 330c to determine pending tasks for bus 330c. In an embodiment, power manager determines to poll bus 330c after memory module 320 finishes performing a task. In another embodiment, the power manager 102 polls the subsystems 304 to determine if they have pending tasks for bus 330c. In step 502, a determination is made regarding whether there are any pending tasks for the bus. For example, power manager 102 can determine whether there are any pending tasks requiring the use of bus 330c. If there are no pending tasks for the bus, the method proceeds to step 504, and a request to power down the bus and/or a subsystem is initiated. For example, power manager 102 can use the method according to FIG. 4 to determine whether bus 330c and/or subsystem 304d should be powered down.

[0061] If there are pending tasks for the bus, the method proceeds to step 506, and a determination is made for the task with the highest precedence. For example, if power manager 102 determines that multiple tasks require the use of bus 330c, then power manager 102 determines which task has the highest precedence. In step 508, a blocker module is configured to recognize a subsystem performing the task as the master of the bus and to block access of other subsystems to the bus. For example, if power manager 102 determines that a task initiated by hub module 308 has the highest precedence, power manager 102 can send a message to LPM 306c instructing LPM 306c to configure blocker module 302c to recognize hub module 308 and/or subsystem 304a as the master of bus 330c. Blocker module 302c responds with an error message if another subsystem attempts to send data over bus 330c while hub module 308 and/or subsystem 304a is the master of bus 330c.

[0062] In step 510, a message is sent to the subsystem informing the subsystem that it is the master of the bus. In an embodiment, step 510 is optional. For example, power manager 102 can send a message to LPM 306a to instruct subsystem 304a and/or hub module 308 that hub module 308 and/or subsystem 304a is the master of bus 330c. In step 512, a message is received from the subsystem indicating that the task is complete. For example, after hub module 308 has finished sending data over bus 330c, hub module 308 can send a message to power manager 102 informing power manager 102 that it has finished its task. In an embodiment, if hub module 308 has no more tasks to perform at this time, hub module 308 can also send a request to power manager 102 for subsystem 304a to be powered down at this time.

[0063] The method can then return back to step 500. For example, after power manager 102 receives the message from hub module 308 indicating that the task is complete, power manager 102 can poll bus 330c again to determine if there are any further tasks pending for bus 330c.

6. EXAMPLE COMPUTER SYSTEM ENVIRONMENT

[0064] It will be apparent to persons skilled in the relevant art(s) that various elements and features of the present disclosure, as described herein, can be implemented in hardware using analog and/or digital circuits, in software, through the execution of instructions by one or more general purpose or special-purpose processors, or as a combination of hardware and software.

[0065] The following description of a general purpose computer system is provided for the sake of completeness. Embodiments of the present disclosure can be implemented in hardware, or as a combination of software and hardware. Consequently, embodiments of the disclosure may be implemented in the environment of a computer system or other processing system. An example of such a computer system 600 is shown in FIG. 6. Modules depicted in FIGS. 1-3 may execute on one or more computer systems 600. Furthermore, each of the steps of the processes depicted in FIGS. 4 and 5 can be implemented on one or more computer systems 600.

[0066] Computer system 600 includes one or more processors, such as processor 604. Processor 604 can be a special purpose or a general purpose digital signal processor. Processor 604 is connected to a communication infrastructure 602 (for example, a bus or network). Various software implementations are described in terms of this exemplary computer system. After reading this description, it will become apparent to a person skilled in the relevant art(s) how to implement the disclosure using other computer systems and/or computer architectures.

[0067] Computer system 600 also includes a main memory 606, preferably random access memory (RAM), and may also include a secondary memory 608. Secondary memory 608 may include, for example, a hard disk drive 610 and/or a removable storage drive 612, representing a floppy disk drive, a magnetic tape drive, an optical disk drive, or the like. Removable storage device 612 reads from and/or writes to a removable storage unit 616 in a well-known manner. Removable storage unit 616 represents a floppy disk, magnetic tape, optical disk, or the like, which is read by and written to by removable storage drive 612. As will be appreciated by persons skilled in the relevant art(s), removable storage unit 616 includes a computer usable storage medium having stored therein computer software and/or data.

[0068] In alternative implementations, secondary memory 608 may include other similar means for allowing computer programs or other instructions to be loaded into computer system 600. Such means may include, for example, a removable storage unit 618 and an interface 614. Examples of such means may include a program cartridge and cartridge interface (such as that found in video game devices), a removable memory chip (such as an EPROM, or PROM) and associated socket, a thumb drive and USB port, and other removable storage units 618 and interfaces 614 which allow software and data to be transferred from removable storage unit 618 to computer system 600.

[0069] Computer system 600 may also include a communications interface 620. Communications interface 620
allows software and data to be transferred between computer system 600 and external devices. Examples of communications interface 620 may include a modem, a network interface (such as an Ethernet card), a communications port, a PCM-CIA slot and card, etc. Software and data transferred via communications interface 620 are in the form of signals which may be electronic, electromagnetic, optical, or other signals capable of being received by communications interface 620. These signals are provided to communications interface 620 via a communications path 622. Communications path 622 carries signals and may be implemented using wire or cable, fiber optics, a phone line, a cellular phone link, an RF link and other communications channels.

As used herein, the terms “computer program medium” and “computer readable medium” are used to generally refer to tangible storage media such as removable storage units 616 and 618 or a hard disk installed in hard disk drive 610. These computer program products are means for providing software to computer system 600.

Computer programs (also called computer control logic) are stored in main memory 606 and/or secondary memory 608. Computer programs may also be received via communications interface 620. Such computer programs, when executed, enable the computer system 600 to implement the present disclosure as discussed herein. In particular, the computer programs, when executed, enable processor 604 to implement the processes of the present disclosure, such as any of the methods described herein. Accordingly, such computer programs represent controllers of the computer system 600. Where the disclosure is implemented using software, the software may be stored in a computer program product and loaded into computer system 600 using removable storage drive 612, interface 614, or communications interface 620.

In another embodiment, features of the disclosure are implemented primarily in hardware using, for example, hardware components such as application-specific integrated circuits (ASICs) and gate arrays. Implementation of a hardware state machine so as to perform the functions described herein will also be apparent to persons skilled in the relevant art(s).

7. CONCLUSION

It is to be appreciated that the Detailed Description, and not the Abstract, is intended to be used to interpret the claims. The Abstract may set forth one or more but not all exemplary embodiments of the present disclosure as contemplated by the inventor(s), and thus, is not intended to limit the present disclosure and the appended claims in any way.

The present disclosure has been described above with the aid of functional building blocks illustrating the implementation of specified functions and relationships thereof. The boundaries of these functional building blocks have been arbitrarily defined herein for the convenience of the description. Alternate boundaries can be defined so long as the specified functions and relationships thereof are appropriately performed.

The foregoing description of the specific embodiments will so fully reveal the general nature of the disclosure that others can, by applying knowledge within the skill of the art, readily modify and/or adapt for various applications such specific embodiments, without undue experimentation, without departing from the general concept of the present disclosure. Therefore, such adaptations and modifications are intended to be within the meaning and range of equivalents of the disclosed embodiments, based on the teaching and guidance presented herein. It is to be understood that the phraseology or terminology herein is for the purpose of description and not of limitation, such that the terminology or phraseology of the present specification is to be interpreted by the skilled artisan in light of the teachings and guidance.

Any representative signal processing functions described herein can be implemented in hardware, software, or some combination thereof. For instance, signal processing functions can be implemented using computer processors, computer logic, application specific circuits (ASIC), digital signal processors, etc., as will be understood by those skilled in the art based on the discussion given herein. Accordingly, any processor that performs the signal processing functions described herein is within the scope and spirit of the present disclosure.

The above systems and methods may be implemented as a computer program executed on a machine, as a computer program product, or as a tangible and/or non-transitory computer-readable medium having stored instructions. For example, the functions described herein could be embodied by computer program instructions that are executed by a computer processor or any one of the hardware devices listed above. The computer program instructions cause the processor to perform the signal processing functions described herein. The computer program instructions (e.g., software) can be stored in a tangible non-transitory computer usable medium, computer program medium, or any storage medium that can be accessed by a computer or processor. Such media include a memory device such as a RAM or ROM, or other type of computer storage medium such as a computer disk or CD ROM. Accordingly, any tangible non-transitory computer storage medium having computer program code that causes a processor to perform the signal processing functions described herein are within the scope and spirit of the present disclosure.

While various embodiments of the present disclosure have been described above, it should be understood that they have been presented by way of example only, and not limitation. It will be apparent to persons skilled in the relevant art that various changes in form and detail can be made therein without departing from the spirit and scope of the disclosure. Thus, the breadth and scope of the present disclosure should not be limited by any of the above-described exemplary embodiments, and further the invention should be defined only in accordance with the following claims and their equivalents.

What is claimed is:

1. A system, comprising:
   a first subsystem comprising a blocker module;
   a bus coupled to the blocker module; and
   a power manager coupled to the first subsystem, wherein
   the power manager is configured to:
   designate a master of the bus, and
   configure the blocker module to:
   detect an attempt to send data over the bus by a second
   subsystem, wherein the second subsystem is not the
   master of the bus,
   block the attempt to send data, and
   send an error message to the second subsystem.

2. The system of claim 1, wherein the first subsystem further comprises a local power manager (LPM), and wherein
the power manager is configured to send a message to the LPM to designate the master and to configure the blocker module.

3. The system of claim 1, wherein the power manager comprises a second blocker module, and wherein the system further comprises:
   - a hub module coupled to the second blocker module and to the first subsystem, wherein the hub module comprises a third blocker module; and
   - a second bus coupled to the third blocker module and to the first subsystem.

4. The system of claim 1, further comprising:
   - a third subsystem coupled to the bus, wherein the third subsystem comprises a second blocker module.

5. The system of claim 1, wherein the power manager is further configured to:
   - poll the bus to determine tasks requesting to use the bus;
   - determine a priority of the tasks; and
   - determine the master based on the priority of the tasks.

6. The system of claim 1, wherein the power manager is further configured to:
   - receive a message from the master indicating that a task is complete; and
   - configure the blocker module to stop blocking data in response to receiving the message.

7. The system of claim 6, wherein the power manager is further configured to:
   - poll the bus in response to receiving the message indicating that the task is complete.

8. The system of claim 6, wherein the power manager is further configured to:
   - determine whether to power down the master in response to receiving the message from the master indicating that the task is complete.

9. The system of claim 1, further comprising:
   - a third subsystem coupled to the bus, wherein the power manager is further configured to:
     - receive a request to power down the third subsystem;
     - determine whether to power down the third subsystem, and
     - in response to determining that the third subsystem should be powered down:
       - configure the blocker module to block data from being sent over the bus, and
       - initiate powering down the third subsystem.

10. The system of claim 9, wherein the power manager is further configured to:
    - poll the bus to determine whether to power down the third subsystem.

11. The system of claim 9, wherein the power manager is further configured to:
    - initiate powering down the bus in response to determining that the third subsystem should be powered down.

12. A system, comprising:
    - a first subsystem comprising a blocker module;
    - a bus coupled to the blocker module;
    - a second subsystem coupled to the bus; and
    - a power manager coupled to the first subsystem, wherein the power manager is configured to:
      - receive a request to power down the second subsystem,
      - determine whether to power down the second subsystem, and
      - in response to determining that the second subsystem should be powered down:
        - configure the blocker module to block data from being sent over the bus, and
        - initiate powering down the second subsystem.

13. The system of claim 12, wherein the power manager is further configured to:
    - poll the bus to determine whether to power down the second subsystem.

14. The system of claim 12, wherein the power manager is further configured to:
    - initiate powering down the bus in response to determining that the second subsystem should be powered down.

15. The system of claim 12, wherein the power manager comprises a second blocker module, and wherein the system further comprises:
    - a second bus coupled to the power manager and to the first subsystem.

16. The system of claim 15, wherein the power manager is further configured to:
    - configure the second blocker module to block traffic over the second bus; and
    - initiate powering down the first subsystem.

17. A method, comprising:
    - determining a priority for tasks requesting to use a bus;
    - designating, based on the priority, a master of the bus;
    - detecting an attempt to send data over the bus by a subsystem, wherein the subsystem is not the master of the bus;
    - blocking the attempt to send data; and
    - sending an error message to the subsystem.

18. The method of claim 17, further comprising:
    - receiving a message from the master indicating that a task is complete; and
    - determining whether to power down the master in response to receiving the message from the master indicating that the task is complete.

19. The method of claim 17, further comprising:
    - in response to determining that the master should be powered down:
      - blocking data traffic to the master, and
      - initiating powering down the master.

20. The method of claim 19, further comprising:
    - initiating powering down a second bus coupled to the master in response to determining that the master should be powered down.