Title: SWITCHING BETWEEN PROCESSOR CACHE AND RANDOM-ACCESS MEMORY

Abstract: The present disclosure describes techniques and apparatuses for switching between processor cache and random-access memory. In some aspects, the techniques and apparatuses are able to reduce die size of application-specific components by forgoing dedicated random-access memory (RAM). Instead of using dedicated RAM, a memory having a cache configuration is reconfigured to a RAM configuration during operations of the application-specific component and then, when the operations are complete, the memory is configured back to the cache configuration. Because many application-specific components already include memory having the cache configuration, reconfiguring this memory rather than including a dedicated RAM reduces die size for the application component.

Fig. 2
Published:

— with international search report (Art. 21(3))
SWITCHING BETWEEN PROCESSOR CACHE AND RANDOM-ACCESS MEMORY

RELATED APPLICATIONS

[0001] This present disclosure claims priority to U.S. Provisional Patent Application Serial No. 61/513,443 filed July 29th, 2011, the disclosure of which is incorporated by reference herein in its entirety.

BACKGROUND

[0002] The background description provided herein is for the purpose of generally presenting the context of the disclosure. Unless otherwise indicated herein, the approaches described in this section are not prior art to the claims in this disclosure and are not admitted to be prior art by inclusion in this section.

[0003] Many conventional system-on-chips (SoCs) include application-specific components, such as application processors and application-specific integrated circuits. These application-specific components are tailored, at least in part, to benefit a specific application, such as through a tailored instruction set or circuitry that provides various functionalities. To continually improve the performance of these application-specific components, more and more functionalities are being added to these application-specific components, thereby increasing die size and production costs.
SUMMARY

[0004] This summary is provided to introduce subject matter that is further described below in the Detailed Description and Drawings. Accordingly, this Summary should not be considered to describe essential features nor used to limit the scope of the claimed subject matter.

[0005] A System-on-Chip is described that comprises a switchable memory configured to switch between a processor cache configuration and a random-access memory (RAM) configuration and a controller. The controller is configured to dynamically switch, responsive to a power-down event or a power-on event, the switchable memory from the processor cache configuration to the RAM configuration, load boot code from a non-volatile memory into the switchable memory while the switchable memory is in the RAM configuration, execute the boot code from the switchable memory while the switchable memory is in the RAM configuration, and dynamically switch, responsive to completion of the execution of the boot code from the switchable memory, the switchable memory from the RAM configuration to the processor cache configuration.

[0006] A method is described comprising, responsive to a power-on event, switching a switchable memory from a processor cache configuration to a random-access memory (RAM) configuration, loading boot code from a non-volatile memory to the switchable memory configured to the RAM configuration, executing a first portion of the boot code from the switchable memory effective to detect and configure a boot device having an operating system image, executing a second portion of the
boot code from the switchable memory effective to boot the operating system image from the boot device, and after booting the operating system image from the boot device, switching the switchable memory from the RAM configuration to the processor cache configuration.

**BRIEF DESCRIPTION OF THE DRAWINGS**

[0007] The details of one or more implementations are set forth in the accompanying figures and the detailed description below. In the figures, the left-most digit of a reference number identifies the figure in which the reference number first appears. The use of the same reference numbers in different instances in the description and the figures indicate like elements.

[0008] Fig. 1 illustrates an operating environment in which techniques for switching between processor cache and random-access memory may be implemented.

[0009] Fig. 2 illustrates a method for switching between processor cache and random-access memory.

[0010] Fig. 3 illustrates a method for booting an operating system, in part by switching a switchable memory from a processor cache configuration to a RAM configuration.

[0011] Fig. 4 illustrates an example applications processor including an example of switchable memory using a switchable level-two (L2) processor cache.

[0012] Fig. 5 illustrates a method for switching from an L2 processor cache configuration to an SRAM configuration.
[0013] Fig. 6 illustrates a method for switching from an SRAM configuration to an L2 processor cache configuration.

[0014] Fig. 7 illustrates a System-on-Chip (SoC) environment for implementing aspects of the techniques described herein.

DETAILED DESCRIPTION

[0015] Overview

This document describes techniques and apparatuses for switching memory between processor cache and random-access memory. These techniques and apparatuses, in some aspects, enable a reduction in die size of application-specific components, which, in many cases, reduces production costs or improves performance. In some aspects, the techniques and apparatuses are able to reduce die size by forgoing dedicated random-access memory (RAM) while still enabling the application-specific components to perform its operations. Instead of using dedicated RAM, a memory having a cache configuration is reconfigured to a RAM configuration during the operations and then, when the operations are complete, is configured back to the cache configuration. Because application-specific components already include memory having the cache configuration, reconfiguring this memory rather than including a dedicated RAM reduces die size for the application component.

[0016] For example, consider an application-specific component tailored to a mobile device, such as a smart-phone that also includes a cellular or baseband
processor. When the smart-phone is first turned on or when resuming from a hibernate or suspend state, an operating system of the smart-phone may be booted or resumed by an application-specific component. This application-specific component conventionally includes a read-only memory (ROM) having boot code (a boot ROM), random-access memory (RAM) into which the boot code is loaded, a boot device having an operating system image, and a processor to execute the boot code once loaded into the RAM. For this application-specific component, the processor loads the boot code into the RAM and then executes the boot code from the RAM to detect the boot device, configure the boot device, authenticate the operating system, and boot the operating system.

[0017] The techniques and apparatuses, however, enable the application-specific component to boot the operating system without using the conventional RAM. Instead, a memory configured to an L2 cache configuration is reconfigured prior to or during the boot process to a static RAM (SRAM) configuration. The boot code is loaded into the memory in the SRAM configuration, which is then executed by the processor to boot the operating system. After the operating system is booted, the memory is switched back to the L2 cache configuration. By so doing, the techniques and apparatuses enable this application-specific component to be built with less memory, which permits a smaller die size and reduces costs.

[0018] The following discussion describes an operating environment, techniques that may be employed in the operating environment, and a System-on-Chip (SoC) in which components of the operating environment can be
embodied. In the discussion below, reference is made to the operating environment by way of example only.

[0019] Operating Environment

Fig. 1 illustrates an example of an operating environment 100 having an application-specific component 102 and a boot device 104, each of which are configured to communicate over data bus 106, such as an integrated-circuit to integrated-circuit (I²C) bus, a low pin count (LPC) bus, a serial peripheral interconnect (SPI) bus, universal asynchronous receiver/transmitter (UART) bus, 1-wire bus, and the like.

[0020] Application-specific component 102 includes one or more microprocessors) 108, switchable memory 110, boot read-only memory (boot ROM) 112, and controller 114. Micro-processor 108 can be any suitable type of processor, either single-core or multi-core, for executing instructions or code associated with programs and/or an operating system of application-specific component 102. Micro-processor 108 may be constructed with or fabricated from any suitable material such as silicon and other semiconductors. Switchable memory 110 is memory configured to being switched between at least two configurations, such as a processor cache configuration and a random-access memory (RAM) configuration. Boot ROM 112 may be configured from any suitable non-volatile memory, such as erasable programmable ROM (EPROM), electronically erasable programmable ROM (EEPROM), and the like. Boot ROM 112 includes boot code that, when executed, aids in booting a system that is associated with application-specific component 102. Note that application-
specific component 102 is tailored, at least in part, to benefit a specific application, in the above-mentioned example to boot an operating system of a smart-phone.

[0021] The techniques and apparatuses, however, are not limited to application-specific components tailored to smart-phones. Instead, various types of application-specific components may be used in conjunction with techniques described herein for deployment in a variety of electronic or computing devices. For example, application-specific components may be configured as integrated-circuits (ASIC), System-on-Chips (SoC), or application-specific standard products (ASSP). Devices in which these application-specific components may be deployed include printers, cameras, copiers, fax machines, household appliances, gaming devices, mobile internet devices, televisions, electronic picture frames, and so on. In some cases an application-specific component may be deployed in multiple such devices with minimal or no reconfiguration.

[0022] Controller 114 is configured to switch switchable memory 110 between configurations, such as by dynamically switching between a processor cache configuration and a random-access memory (RAM) configuration. Controller 114 may be embodied in various manners, such as executable code stored in boot ROM 112 for execution by micro-processors 108, a state machine, hardware (e.g., logic circuitry), firmware, or any suitable combination thereof. How controller 114 is used and implemented varies, and is described in more detail below.

[0023] Boot device 104 stores operational firmware or an operating system of application-specific component 102. In this particular example, boot device 104
stores operating system image 116. Boot device 104 may be configured from any suitable type of hardware memory or memory device, such as non-volatile RAM (NVRAM), EEPROM, flash memory, and the like. Firmware or operating systems may be stored by boot device 104 in various manners such as an operating system image, firmware modules, microcode, and so on. Alternately or additionally, contents of boot device 104 can be authenticated as secure or trusted code prior to execution in secure-execution environments. Although shown in example environment 100 as separate entities, boot device 104 and application-specific component 102 may be physically integrated in other aspects of the techniques and apparatuses disclosed herein.

[0024] **Switching Between Processor Cache and Random-Access Memory**

The following discussion describes techniques for switching a memory between processor-cache and random-access memory (RAM) configurations. These techniques can be implemented using the previously described environments, such as controller 114 of Fig. 1. These techniques include methods illustrated in Figs. 2, 3, 5, and/or 6, each of which is shown as a set of operations performed by one or more entities. These methods are not necessarily limited to the orders shown for performing the operations. Further, these methods may be used in conjunction with one another, in whole or in part, whether performed by the same entity, separate entities, or any combination thereof. In portions of the following discussion, reference will be made to operating environment 100 of Fig. 1 by way of example.
Such reference is not to be taken as limited to operating environment 100 but rather as illustrative of one of a variety of examples.

[0025] Fig. 2 illustrates a method 200 for switching between processor cache and random-access memory. At 202, a switchable memory is dynamically switched, responsive to a power-down event or a power-on event, from a processor cache configuration to a random-access memory (RAM) configuration. A power-down event may include a soft or hard reset, entering a low-power state (e.g., suspend or hibernation modes), complete system shut down, and so on. A power-on event may include a power-on reset (hard or soft), re-boot, cold boot (e.g., system-start from an off state), warm boot, resume from a suspend or hibernation mode, and so on. Consider a case where controller 114 of Fig. 1 receives an indication that a power-down event is occurring or is about to occur. Assume that this power-down event indicates that a computing system (e.g., a desktop computer, smart-phone, laptop computer, tablet computer, gaming system, etc.) is, or will soon be entering, a low-power or unpowered state, such as a hibernate or suspend state. In this case, controller 114, prior to entering the hibernate or suspend state, switches switchable memory 110 from a current processor cache configuration to a RAM configuration in preparation for a later power-on event in which the computing system will be booted.

[0026] Controller 114 may instead act responsive to a power-on event, such as a reset or a system-start, to switch the switchable memory to the RAM configuration. In this case, controller 114 dynamically switches switchable memory 110 from an existing processor cache configuration to a RAM configuration in preparation for
executing boot code from boot ROM 112. Various manners in which controller 114 may act to switch between configurations, such as a level-two processor cache configuration (L2 configuration) to a static RAM (SRAM) configuration, are set forth in detail as part of methods 500 and 600 of Figs. 5 and 6 below.

[0027] At 204, boot code is loaded from a non-volatile memory into the switchable memory while the switchable memory is in the RAM configuration. The non-volatile memory may be a boot ROM configured to store the boot code as described above. Controller 114, for example, loads boot code from boot ROM 112 into switchable memory 110 from boot ROM 112 effective to enable execution of the boot code by micro-processor 108. In some cases, the operating system image may be stored on boot device 104 in flash memory. In such a case, controller 114 may use a flash controller to load operating system image 116. As part of booting the computing system, controller 114 may also authenticate operating system image 116.

[0028] At 206, boot code is executed from the switchable memory while the switchable memory is configured in the RAM configuration. Execution of the boot code may be effective to boot an operating system or firmware of an application-specific component. The operating system or firmware may be stored in a boot device, such as a flash or EEPROM module internal to or external to the application-specific component. In some cases, the operating system or firmware may be authenticated as trusted or secure prior to booting. Continuing the ongoing example, micro-processor 108 executes boot code from switchable memory 110 to authenticate and boot operating system image 116 from boot device 104.
At 208, responsive to booting the operating system, the switchable memory is dynamically switched from the RAM configuration to the processor cache configuration. In so doing, controller 114 may enable the switchable memory to be used in its prior configuration. For example, while an operating system is being executed, microprocessor 108 may use switchable memory 110 as a cache for data or instructions associated with execution of the operating system or other programs.

Concluding the present example, micro-processors 108 accesses data of switchable memory 110 in the processor cache configuration. This is illustrated at 210, where, after dynamically switching the switchable memory from the RAM configuration to the processor cache configuration, the switchable memory is enabled for use in the processor cache configuration. As noted above, various manners in which controller 114 may act to switch between configurations are set forth in detail below.

Fig. 3 illustrates a method 300 for booting an operating system, in part by switching a switchable memory from a processor cache configuration to a RAM configuration. In portions of the following discussion, reference will be made to operating environment 100 of Fig. 1 and entities of Fig. 4 by way of example. Such reference is not to be taken as limited to Figs. 1 or 4, but rather as illustrative of one of a variety of examples.

At 302, responsive to a power-on event, switchable memory is switched from a processor cache configuration to a RAM configuration. This power-on event
may be a reset, cold boot, or a resume from a low-power mode, such as a sleep or hibernation state.

[0033] By way of example, consider method 300 in the context of application processor 402 of Fig. 4, which is an example of application-specific component 102 of Fig. 1. Application processor 402 includes boot ROM 112 and controller 114 both of Fig. 1, as well as an example of switchable memory 110, here switchable level-two (L2) processor cache 404. Application processor 402 also includes a double data-rate synchronous DRAM controller 406 (DDR controller 406), a level-one (L1) execution translation lookaside buffer (ITLB or I for short) processor cache 408, a level-one (L1) read/write translation lookaside buffer (DTLB or D for short) cache 410, and a flash controller 412. Flash controller 412 is configured to control external flash 414. DDR controller 406 is configured to controlling double data-rate synchronous DRAM 416 (DDR memory 416).

[0034] Controller 114 is configured to communicate with and/or be executed using, in whole or in part, cellular processor 418. At 302, a power-on event is detected, such as by a power-management subsystem (not shown) of application processor 402. Here, L2 cache 404 is configured such that its configuration can be altered, including dynamically during a boot process (or conversely during a power-down sequence). At 302, controller 114 switches L2 cache 404, which is configured as processor cache, to a static random-access memory (SRAM) configuration. While the methods are not required to switch this configuration in a particular manner, consider method 500, which provide a detailed embodiment showing how this
switching of configuration from an L2 cache configuration to an SRAM configuration may be performed. Method 500 is described in detail below.

[0035] At 304, boot code is loaded from a non-volatile memory device into the switchable memory configured to the RAM configuration. Here the boot code is loaded from boot ROM 112 into L2 processor cache 404 of application processor 402.

[0036] At 306, a first portion of the boot code is executed effective to detect and configure a boot device. This boot device may store an operating system, an operating system image, or firmware for an application-specific component. The boot device may be configured from any suitable type of memory or memory module, such as an EEPROM or flash memory. Continuing the ongoing embodiment, application processor 402 executes the boot code from L2 cache 404 effective to detect and configure external flash 414. Here, flash controller 412 is used in the detection and configuration of external flash 414, as application processor 402, as presently configured, may not directly access flash memory 414.

[0037] Optionally at 308, a second portion of boot code is executed effective to authenticate an operating system image of the boot device as trusted or secure. In some cases, execution or booting of the contents of the boot device may be prevented when authentication fails. In other cases, booting or execution may proceed, while an indication of the failed authentication is presented to a user or reported to a security policy manager. In the context of the present example, application processor 402 executes additional boot code to authenticate an operating system image stored in external flash 414.
At 310, a third portion of the boot code is executed effective to boot the operating system image from the boot device. The boot code, as noted, includes at least some code effective to boot the operating system, though other code, such as executable instructions to perform operations of controller 114, also may be included. In the ongoing example where the operating system image is stored on external flash 414, executing the ROM code from the switchable memory is effective to boot the operating system image from external flash 414 via flash controller 412.

At 312, after the operating system image is booted from the boot device, the switchable memory is switched from the SRAM configuration to the processor cache configuration. This permits other uses of the switchable memory, here permitting the switchable memory to be used as processor cache. Concluding the present example, controller 114 configures L2 cache 404 from the SRAM configuration used to boot the operating system of application processor 402 back to an L2 cache configuration. Consider method 600 as described below, which provides a detailed embodiment showing how this switching of configuration back from an SRAM configuration to an L2 cache configuration may be performed.

Note that the techniques, whether according to method 200, method 300, or a combination thereof, in whole or in part, may perform a trusted or untrusted boot of a computing system (e.g., operation 308).

Fig. 5 illustrates a method 500 for switching from an L2 processor cache configuration to an SRAM configuration. Method 500 is but one example way in which the techniques may switch configurations of a switchable memory. As noted,
in some cases this permits lower-costs or higher performance by reducing memory on
an application-specific component.

[0042] At 502, an enable bit of a control register of a switchable memory is
written to disable the switchable memory as L2 cache. At 504, the switchable
memory is cleaned using a "clean all" instruction. At 506, a select bit in a
configuration register of the switchable memory is cleared. At 508, an "invalidate
all" instruction is used to invalidate the switchable memory as L2 cache. At 510,
methods 500 wait for the switchable memory to become idle. At 512, an SRAM
select bit in the configuration register of the switchable memory is set. At 514, a bank
of the switchable memory is configured for SRAM.

[0043] Fig. 6 depicts a method 600 for switching from an SRAM configuration
to an L2 processor cache configuration. Method 600 is but one example way in which
the techniques may switch configurations of a switchable memory.

[0044] At 602, data from a bank of the switchable memory is copied to another
memory. Thus, data in the bank of L2 cache 404 configured for SRAM is copied to
another memory, such as one external to application processor 402, such as a DDR
416 through DDR controller 406 or external flash 414 through flash controller 412.
At 604, the bank of the switchable memory is disabled. At 606, an L2 select bit is
cleared. At 608, an L2 cache enable bit in a control register is cleared. At 610, the
switchable memory is invalidated using an L2 cache "invalidate all" instruction.

[0045] While methods 500 and 600 are performed through particular bits in
various registers associated with the switchable memory and using particular
instructions, these are intended as examples and not to limit methods 500 or 600, or methods 200 or 300, to these illustrative examples. Alternate aspects are also contemplated, such as the use of logic circuitry or a state machine associated with monitoring and/or controlling power state transitions of an application-specific component.

[0046] **System-on-Chip**

Fig. 7 illustrates a System-on-Chip (SoC) 700, which can implement various aspects described above. An SoC can be implemented in any suitable device, such as a video game console, IP-enabled television, desktop computer, laptop computer, tablet computer, smart-phone, server, network-enabled printer, set-top box, printer, scanner, camera, picture frame, and/or mobile internet device.

[0047] SoC 700 can be integrated with electronic circuitry, a microprocessor, memory, input-output (I/O) logic control, communication interfaces and components, other hardware, firmware, and/or software. SoC 700 can also include an integrated data bus (not shown) that couples the various components of the SoC for data communication between the components.

[0048] In this example, SoC 700 includes various components such as an input-output (I/O) logic control 702 (e.g., to include electronic circuitry) and a microprocessor 704 (e.g., any of a microcontroller or digital signal processor). SoC 700 also includes a memory 706, which can be any type of RAM, low-latency nonvolatile memory (e.g., flash memory), ROM, and/or other suitable electronic data storage. SoC 700 can also include various firmware and/or software, such as an
operating system 708, which can be computer-executable instructions maintained by memory 706 and executed by microprocessor 704. SoC 700 can also include other various communication interfaces and components, communication components, other hardware, firmware, and/or software.

[0049] SoC 700 includes switchable memory 110, boot ROM 112, and controller 114. Examples of these various components, functions, and/or entities, and their corresponding functionality, are described with reference to the respective components of the environment 100 shown in Fig. 1 and Fig. 4. Controller 114 and the other components can be implemented as hardware, firmware, fixed logic circuitry, or any combination thereof that is implemented in connection with the I/O logic control 702 and/or other signal processing and control circuits of SoC 700.

[0050] Although the subject matter has been described in language specific to structural features and/or methodological operations, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the specific features or operations described above, including orders in which they are performed.
CLAIMS

What is claimed is:

1. A System-on-Chip (SoC) comprising:
   a switchable memory configured to switch between a processor cache configuration and a random-access memory (RAM) configuration; and
   a controller configured to:
      dynamically switch, responsive to a power-down event or a power-on event, the switchable memory from the processor cache configuration to the RAM configuration;
      load boot code from a non-volatile memory into the switchable memory while the switchable memory is in the RAM configuration;
      execute the boot code from the switchable memory while the switchable memory is in the RAM configuration effective to boot an operating system of the SoC; and
      dynamically switch, responsive to booting the operating system of the SoC, the switchable memory from the RAM configuration to the processor cache configuration.

2. The System-on-Chip of claim 1, wherein the non-volatile memory is a boot read-only memory (ROM) within the SoC that is configured to store the boot code.
3. The System-on-Chip of claim 2, wherein the SoC does not include a random-access memory dedicated for execution of the boot code.

4. The System-on-Chip of claim 1, wherein the controller is further configured to dynamically switch the switchable memory from the processor cache configuration to the RAM configuration responsive to the power-down event and prior to the power-on event, the power-down event being a hibernation or suspend mode, the power-on event being a reset or cold boot.

5. The System-on-Chip of claim 1, wherein the controller is further configured to dynamically switch the switchable memory from the processor cache configuration to the RAM configuration responsive to the power-on event, the power-on event being a reset from a hibernation mode or a suspend mode.

6. The System-on-Chip of claim 1, wherein executing the boot code is effective to boot the operating system of the SoC from a boot device, the boot device configured to store an operating system image of the operating system.

7. The System-on-Chip of claim 6, wherein the SoC further comprises a flash controller, the boot device comprises flash memory, and wherein booting the operating system of the SoC includes using the flash controller to access the operating system image from the flash memory.
8. The System-on-Chip of claim 1, wherein the controller is further configured, after dynamically switching the switchable memory from the RAM configuration to the processor cache configuration, to use or enable use of the switchable memory while the switchable memory is in the processor cache configuration.

9. The System-on-Chip of claim 1, wherein the processor cache configuration is a second-level (L2) configuration and the RAM configuration is a static RAM (SRAM) configuration and the controller is further configured to dynamically switch the memory from the L2 cache configuration to the SRAM configuration by:

   disabling the switchable memory as L2 cache by writing to an enable bit in a control register;

   cleaning the switchable memory;

   clearing a read-only memory (RAM) select bit in a configuration register;

   invalidating the switchable memory as L2 cache;

   waiting for the switchable memory to become idle;

   setting the RAM select bit in the configuration register; and

   configuring a bank of the switchable memory for SRAM.
10. The System-on-Chip of claim 1, wherein the processor cache configuration is a second-level (L2) configuration and the RAM configuration is a static RAM (SRAM) configuration and the controller is further configured to dynamically switch the memory from the SRAM configuration to the L2 cache configuration by:

- copying data from a bank of the switchable memory to a second memory;
- disabling the bank of the switchable memory;
- clearing an L2 select bit;
- clearing the L2 cache enable bit in a control register; and
- invalidating the switchable memory.

11. The System-on-Chip of claim 10, wherein the second memory is a dynamic RAM (DRAM) or flash memory external to the SoC.
12. A method comprising:

responsive to a power-on event, switching a switchable memory from a processor cache configuration to a random-access memory (RAM) configuration;

loading boot code from a non-volatile memory to the switchable memory configured to the RAM configuration;

executing a first portion of the boot code from the switchable memory effective to detect and configure a boot device having an operating system image;

executing a second portion of the boot code from the switchable memory effective to boot the operating system image from the boot device; and

after booting the operating system image from the boot device, switching the switchable memory from the RAM configuration to the processor cache configuration.

13. The method of claim 12, wherein the power-on event is a reset, cold boot, resume from a hibernate mode, or resume from a sleep mode.

14. The method of claim 12, wherein the processor cache configuration is a second-level (L2) cache configuration and the RAM configuration is a static RAM (SRAM) configuration.
15. The method of claim 14, wherein switching the switchable memory from the processor cache configuration to the RAM configuration comprises:

- disabling the switchable memory as L2 cache by writing to an enable bit in a control register;
- cleaning the switchable memory;
- clearing a RAM select bit in a configuration register;
- invalidating the switchable memory as L2 cache;
- waiting for the switchable memory to become idle;
- setting the RAM select bit in the configuration register; and
- configuring a bank of the switchable memory for SRAM.

16. The method of claim 14, wherein switching the switchable memory from the RAM configuration to the processor cache configuration comprises:

- copying data from a bank of the switchable memory to a second memory;
- disabling the bank of the switchable memory;
- clearing an L2 select bit;
- clearing an L2 cache enable bit in a control register; and
- invalidating the switchable memory.

17. The method of claim 16, wherein the second memory is a dynamic RAM (DRAM) or flash memory external to a System-on-Chip within which the switchable memory is integrated.
18. The method of claim 12, wherein the method is performed by an application specific integrated-circuit (ASIC), a System-on-Chip (SoC), an application processor, or a cellular processor.

19. The method of claim 12, wherein the boot device includes flash memory and executing the second portion of the boot code from the switchable memory is effective to boot the operating system image from the boot device using a flash controller.

20. The method of claim 12 further comprising, prior to booting the operating system image, executing another portion of the boot code from the switchable memory effective to authenticate the operating system image as a trusted operating system image.
Fig. 1
Dynamically Switch, Responsive to a Power-Down Event or a Power-On Event, a Switchable Memory from a Processor Cache Configuration to a RAM Configuration

Load Boot Code from Boot ROM into the Switchable Memory while the Switchable Memory is in the RAM Configuration

Execute the Boot Code from the Switchable Memory to Boot an Operating System

Dynamically Switch, Responsive to Booting the Operating System, the Switchable Memory from the RAM Configuration to the Processor Cache Configuration

Enable Use of the Switchable Memory while in Processor Cache Configuration

Fig. 2
Responsive to a Power-On Event, Switch a Switchable Memory from a Processor Cache Configuration to a Random-Access Memory (SRAM) Configuration

Load Boot Code From a Non-Volatile Memory to the Switchable Memory Configured to the SRAM Configuration

Execute a First Portion of Boot Code to Detect and Configure a Boot Device having an Operating System Image

Execute a Second Portion of the Boot Code to Authenticate the Operating System Image of the Boot Device

Execute a Third Portion of the Boot Code to Boot the Operating System Image from the Boot Device

After Booting the Operating System Image from the Boot Device, Switch the Switchable Memory from the SRAM Configuration to the Processor Cache Configuration

Fig. 3
Disable Switchable Memory as L2 Cache by Writing to an Enable Bit in a Control Register

Clean Switchable Memory Using a Clean All Instruction

Clear RAM Select Bit in Configuration Register

Invalidate Switchable Memory as L2 Cache using Invalidate All Instruction

Wait for Switchable Memory to Become Idle

Set RAM Select Bit in Configuration Register

Configure a Bank of the Switchable Memory for SRAM
Copy Data from Bank of Switchable Memory to Another Memory 602

Disable the Bank of the Switchable Memory 604

Clear L2 Select Bit 606

Clear L2 Cache Enable Bit in a Control Register 608

Invalidate Switchable Memory Using L2 Cache Invalidate All Instruction 610
INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F9/44
ADD.

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

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

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>paragraph [0018] - paragraph [0029] ---------------------------------------------</td>
<td></td>
</tr>
</tbody>
</table>

Further documents are listed in the continuation of Box C.

* 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
* "A" document member of the same patent family

Date of the actual completion of the international search 12 October 2012

Date of mailing of the international search report 19/10/2012

Name and mailing address of the ISA/Authorized officer
European Patent Office, P.B. 5818 Patentilaan 2
NL-2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Ebert, Werner

See patent family annex.
<table>
<thead>
<tr>
<th>Patent document cited in search report</th>
<th>Publication date</th>
<th>Patent family member(s)</th>
<th>Publication date</th>
</tr>
</thead>
<tbody>
<tr>
<td>US 2008005549 Al</td>
<td>03-01-2008</td>
<td>CN 101097522 A</td>
<td>02-01-2008</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP 2008016020 A</td>
<td>24-01-2008</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 2008005549 Al</td>
<td>03-01-2008</td>
</tr>
</tbody>
</table>