(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
18 March 2010 (18.03.2010)

PCT

(10) International Publication Number
WO 2010/029009 A1

(51) International Patent Classification:
G06F 12/08 (2006.01) *G06F 15/163* (2006.01)
G06F 13/364 (2006.01)

(21) International Application Number:
PCT/EP2009/061396

(22) International Filing Date:
3 September 2009 (03.09.2009)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
12/208,651 11 September 2008 (11.09.2008) US

(71) Applicant *(for all designated States except US)*: **INTERNATIONAL BUSINESS MACHINES CORPORATION** [US/US]; New Orchard Road, Armonk, New York 10504 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **DUVALSAINT, Karl** [US/US]; IBM Corporation, 2455 South Road, MP328, Poughkeepsie, New York 12601-5400 (US). **KIM, Daeik** [KR/US]; IBM Corporation, 2070 Route 52, MP27A, Hopewell Junction, NY 12533-6683 (US). **HOFSTEE, Harm, Peter** [NL/US]; IBM Corporation, 11501 Burnet Road, MP9062007D, Austin, Texas 78758-3400 (US). **KIM, Moon, Ju** [US/US]; IBM Corporation, Dept IQ0A/Bldg 040-3, 1701 North Street, Endicott, New York 13760 (US).

(74) Agent: **LING, Christopher, John**; IBM United Kingdom Limited, Intellectual Property Law, Hursley Park, Winchester Hampshire SO21 2JN (GB).

(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, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

[Continued on next page]

(54) Title: VIRTUALIZATION IN A MULTI-CORE PROCESSOR (MCP)

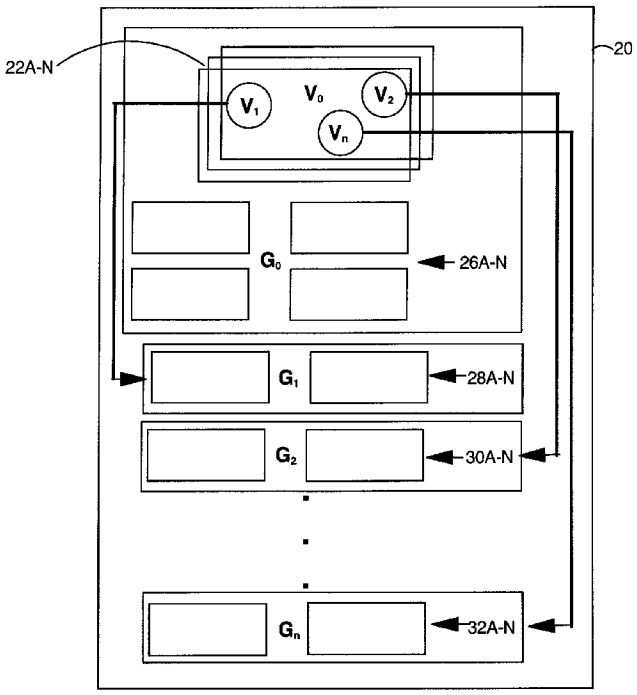

FIG. 2

(57) Abstract: This invention describes an apparatus, computer architecture, method, operating system, compiler, and application program products for MPEs as well as virtualization in a symmetric MCP. The disclosure is applied to a generic microprocessor architecture with a set (e.g., one or more) of controlling elements (e.g., MPEs) and a set of groups of sub- processing elements (e.g., SPEs). Under this arrangement, MPEs and SPEs are organized in a way that a smaller number MPEs control the behavior of a group of SPEs. The apparatus enables virtualized control threads within MPEs to be assigned to different groups of SPEs for controlling the same. The apparatus further includes a MCP coupled to a power supply coupled with cores to provide a supply voltage to each core (or core group) and controlling- digital elements and multiple instances of sub-processing elements.

WO 2010/029009 A1

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (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, 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))*

VIRTUALIZATION IN A MULTI-CORE PROCESSOR (MCP)

FIELD OF THE INVENTION

The present invention generally relates to virtualization in a (e.g., symmetric) multi-core processor (MCP). Specifically, the present invention associates/assigns virtualization threads of a main processing element (MPE) to groups of sub-processing elements (SPEs) for improved operation management and power savings.

BACKGROUND OF THE INVENTION

Low utilization of Multi-Core Processors (MCPs) has been a major drawback of symmetric MCPs. Also, design inflexibility forces continuous leakage current in the unloaded and stand-by sub-elements, such as Sub-Processing Element (SPE), so that the power is wasted. For example, in a symmetric MCP, there can be a Main Processing Element (MPE) and 8 SPEs. In many cases, only a portion of SPEs are utilized and the overall MCP utilization is usually low. Such stand-by SPEs consume high levels of power and continuously leak. Typically, a MCP is used for the high performance digital processor scaling, but due to the complexity of the MCP design, the utilization and the efficiency of the software become challenging to optimize as the MCP dimension increases.

SUMMARY OF THE INVENTION

The present invention accordingly provides, in a first aspect, a multi-core processor, comprising: a set of main processing elements each comprising a set of virtualized control threads; and a set of groups of sub-processing elements, each one of the set of virtualized control threads controlling at least one of the set of groups of sub-processing elements.

Preferably, each of the set of groups of sub-processing elements comprises a plurality of sub-processing elements. Preferably, the set of virtualized control threads is configured to send program code and data to the set of groups of sub-processing elements. Preferably, the set of virtualized control threads is further configured to collect computation results from the set of

groups of sub-processing elements. Preferably, the set of virtualized control threads controls a clock speed, power consumption and computation loading of the set of groups of sub-processing elements. Preferably, the set of virtualized control threads is configured to send and receive power control requests between the set of main processing elements and the set of groups of sub-processing elements. Preferably, the set of main processing elements comprises a plurality of main processing elements. Preferably, each one of the set of visualized control threads is associated with a single group of sub-processing elements.

In a second aspect, there is provided a processing system, comprising: a main processing element; a group of sub-processing elements; and a virtualized control thread associating the main processing element with the group of sub-processing elements, the virtualized control thread controlling the group of sub-processing elements.

Preferably, the virtualized control thread is embodied within the set of main processing elements. Preferably, the virtualized control thread is configured to send program code and data to the group of sub-processing elements. Preferably, the virtualized control thread is further configured to collect computation results from the group of sub-processing elements.

Preferably, the virtualized control thread controls a clock speed, power consumption and computation loading of the group of sub-processing elements.

In a third aspect, there is provided a processing method, comprising: associating a set of main processing elements with a set of groups of sub-processing elements using a set of virtualized control threads; and controlling the set of groups of sub-processing elements using the set of virtualized control threads.

The processing method may further comprise sending program code and data to the set of groups of sub-processing elements via the set of virtualized control threads. The processing method may further comprise receiving computation results from the set of groups of sub-processing elements via the set of virtualized control threads. Preferably, the controlling comprises controlling a clock speed, a power consumption and a computation loading of the set of groups of sub-processing elements. Preferably, the set of virtualized control threads is

embodied as program code within the set of main processing elements. Preferably, the set of main processing elements, the set of groups of sub-processing elements, and the set of virtualized control threads comprises a multi-core processor.

In a fourth aspect, there is provided a method for deploying a processing system, comprising: providing a multi-core processor comprising: a set of main processing elements each comprising a set of virtualized control threads; and a set of groups of sub-processing elements, each one of the set of virtualized control threads controlling at least one of the set of groups of sub-processing elements.

This invention thus preferably provides an apparatus, computer architecture, method, operating system, compiler, and application program products for MPEs as well as virtualization in a symmetric MCP. The disclosure is applied to a generic microprocessor architecture with a set (e.g., one or more) of controlling elements (e.g., MPEs) and a set of groups of sub-processing elements (e.g., SPEs). Under this arrangement, MPEs and SPEs are organized in a way that a smaller number MPEs control the behavior of a group of SPEs. The apparatus enables virtualized control threads within MPEs to be assigned to different groups of SPEs for controlling the same. The apparatus further includes a MCP coupled to a power supply coupled with cores to provide a supply voltage to each core (or core group) and controlling-digital elements and multiple instances of sub-processing elements.

The first aspect of the present invention thus preferably provides a multi-core processor, comprising: a set of main processing elements each comprising a set of virtualized control threads; and a set of groups of sub-processing elements, each one of the set of virtualized control threads controlling at least one of the set of groups of sub-processing elements.

The second aspect of the present invention preferably provides a processing system, comprising: a main processing element; a group of sub-processing elements; and a virtualized control thread associating the main processing element with the group of sub-processing elements, the virtualized control thread controlling the group of sub-processing elements.

The third aspect of the present invention preferably provides a processing method, comprising: associating a set of main processing elements with a set of groups of sub-processing elements using a set of virtualized control threads; and controlling the set of groups of sub-processing elements using the set of virtualized control threads.

The fourth aspect of the present invention preferably provides a method for deploying a processing system, comprising: providing a multi-core processor comprising: a set of main processing elements each comprising a set of virtualized control threads; and a set of groups of sub-processing elements, each one of the set of virtualized control threads controlling at least one of the set of groups of sub-processing elements.

The fifth aspect of the present invention preferably provides a computer-implemented method, comprising: associating a set of main processing elements with a set of groups of sub-processing elements using a set of virtualized control threads; and controlling the set of groups of sub-processing elements using the set of virtualized control threads.

BRIEF DESCRIPTION OF THE DRAWINGS

A preferred embodiment of the present invention will now be described, by way of example only, with reference to the accompanying drawings, in which:

Fig. 1 shows a related art processor.
Fig. 2 shows multi-core processor according to the present invention.

The drawings are not necessarily to scale. The drawings are merely schematic representations, not intended to portray specific parameters of the invention. The drawings are intended to depict only typical embodiments of the invention, and therefore should not be considered as limiting the scope of the invention. In the drawings, like numbering represents like elements.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

As indicated above, this invention describes an apparatus, computer architecture, method, operating system, compiler, and application program products for MPEs as well as virtualization in a symmetric MCP. The disclosure is applied to a generic microprocessor architecture with a set (e.g., one or more) of controlling elements (e.g., MPEs) and a set of groups of sub-processing elements (e.g., SPEs). Under this arrangement, MPEs and SPEs are organized in a way that a smaller number MPEs control the behavior of a group of SPEs. The apparatus enables virtualized control threads within MPEs to be assigned to different groups of SPEs for controlling the same. The apparatus further includes a MCP coupled to a power supply coupled with cores to provide a supply voltage to each core (or core group) and controlling-digital elements and multiple instances of sub-processing elements.

Under related art systems, such as that shown in Fig. 1, the MCP 10 provides a conventional operation: a MPE 12 controls a group of SPEs 14A-N. As only a portion of SPEs are utilized, the overall MCP utilization is usually low. Moreover, stand-by SPEs 16A-N consume high levels of power and continuously leak. Typically, a MCP is used for the high performance digital processor scaling, but due to the complexity of the MCP design, the utilization and the efficiency of the software become challenging to optimize as the MCP dimension increases.

To address these issues, a configuration such as that shown in Fig. 2 is provided. As depicted, MCP 20 comprises a set of MPEs 22A-N. Each MPE 22A-N comprises a set of virtualized control threads V1-VN. Under the present invention MPEs 22A-N allow multiple virtualized control threads V1-VN that are each assigned to/associated with a (e.g., single) group of SPEs. For example, as further shown in Fig. 1, MCP 20 further includes groups G0-Gn of SPEs. In the example shown, group G0 comprises SPEs 26A-N, group G1 comprises SPEs 28A-N, group G2 comprises SPEs 30A-N, and group Gn comprises SPEs 32A-N. It should be understood that the configuration shown is for illustrative purposes only and neither the quantity of SPEs nor their grouping is intended to limit the teachings recited herein. In any event, as can been seen, virtualized control threads V1-Vn are each associated with a specific group of SPEs. That is, virtualized control thread V1 is associated

with SPE group G1, virtualized control thread V2 is associated with SPE group G2, while virtualized control thread Vn is associated with SPE group Gn.

In a typical embodiment, there is a one-to-one relationship between threads and groups (however, this need not be the case in that a single virtualized control thread could be associated with multiple groups). In addition, all virtualized control threads V1-Vn are shown as being embodied within a single, common MPR 22. However, it should be understood that this need not be the case. For example, virtualized control threads could be embodied in two or more different MPEs 22A-N.

Regardless, each virtualized control thread V1-Vn serves to associate/link at least one MPE with at least one group of SPEs. This allows the SPEs to be controlled. Specifically, each virtualized control thread V1-Vn is typically implemented as computer program code (e.g., contained on a computer readable medium). The virtualized control threads have several functions including (among others) the following: sending program code and data to the set of groups of sub-processing elements; collecting computation results from the set of groups of sub-processing elements; sending and receiving power control requests between the set of main processing elements and the set of groups of sub-processing element; and controlling a clock speed, power consumption and computation loading of the set of groups of sub-processing elements.

A more specific illustrative embodiment of the association of virtualized control threads with groups of SPEs will be given below:

Using the teachings recited above, each MPE allows additional virtualizations, based upon the SPEs availability and the number of requested SPEs. The MPE keeps a log of which SPEs "belong" to the MPE. The computation-loaded SPEs as well as unloaded & free SPEs are accounted within the log. The virtualization (e.g., virtualized control threads) is initiated by either a software or hardware request. An Operating System (OS) is designed to enable multiple independent threads with virtualization. Computer programming languages and the machine code compilers support the virtualization with libraries. The MPE also can tune the rest of the virtualized control threads using time-division resource sharing and OS-level

preemptive task management to allow for multi-tasking and virtualization. By allowing main and pseudo MPEs, the MPE loading is reduced and becomes more efficient. Therefore the MCP performance vs. power efficiency increases.

As described above in conjunction with Fig. 2, a group of SPEs (Gn) are assigned to the virtualized control threads. The MPE maintains different virtualized control threads over the SPEs, as shown in the diagram. The virtualized control thread loads executable program codes and data to its associated SPE group, and monitors the progress and controls the performance and power consumption of the SPE group. When the SPE group produces computation results, the virtualized control thread sends the results to the MPE, allowing further computations or external input/output (I/O). The number of total virtualization is limited by the MPE and SPE capacity to hold virtualization, and the number of available SPEs. Further requests for virtualization can be denied, or accepted by sharing time-division thread sharing.

The virtualized control threads in the MPE control the power supply voltage and clock frequency to each SPE group, so that the active and stand-by currents are optimized for the required computations. Typically, a digital circuit speed increases when the supply voltage is raised, and the clock speed can be increased. Within the allowed supply voltage range, it is adjusted based on the computation requirements. When the loaded computation requires intensive operation, the voltage and clock frequency are raised so that it is completed within the time frame. When the loaded computation is loose and if there are plenty of time for processing, the voltage and the clock frequency are lowered to maximize performance/power ratio. An extreme case is when SPEs are standing by. The supply voltage can be reduced to 0 voltage to put the SPE is sleep mode. Slightly higher voltages can be used with a trade-off between leakage current and wake up time. It takes more time to wake an element when it is in a deeper sleep mode.

It should be understood that the present invention could be deployed on one or more computing devices (e.g., servers, clients, etc.) within a computer infrastructure. This is intended to demonstrate, among other things, that the present invention could be implemented within a network environment (e.g., the Internet, a wide area network (WAN),

a local area network (LAN), a virtual private network (VPN), etc.), or on a stand-alone computer system. In the case of the former, communication throughout the network can occur via any combination of various types of communications links. For example, the communication links can comprise addressable connections that may utilize any combination of wired and/or wireless transmission methods. Where communications occur via the Internet, connectivity could be provided by conventional TCP/IP sockets- based protocol, and an Internet service provider could be used to establish connectivity to the Internet. Still yet, the computer infrastructure is intended to demonstrate that some or all of the components of such an implementation could be deployed, managed, serviced, etc. by a service provider who offers to implement, deploy, and/or perform the functions of the present invention for others.

Where computer hardware is provided, it is understood that any computers utilized will include standard elements such as a processing unit, a memory medium, a bus, and input/output (I/O) interfaces. Further, such computer systems can be in communication with external I/O devices/resources. In general, processing units execute computer program code, such as the functionality described above (e.g., all libraries discussed herein), which is stored within memory medium(s). While executing computer program code, the processing unit can read and/or write data to/from memory, I/O interfaces, etc. The bus provides a communication link between each of the components in a computer. External devices can comprise any device (e.g., keyboard, pointing device, display, etc.) that enable a user to interact with the computer system and/or any devices (e.g., network card, modem, etc.) that enable the computer to communicate with one or more other computing devices.

The hardware used to implement the present invention can comprise any specific purpose computing article of manufacture comprising hardware and/or computer program code for performing specific functions, any computing article of manufacture that comprises a combination of specific purpose and general purpose hardware/software, or the like. In each case, the program code and hardware can be created using standard programming and engineering techniques, respectively. Moreover, the processing unit therein may comprise a single processing unit, or be distributed across one or more processing units in one or more locations, e.g., on a client and server. Similarly, the memory medium can comprise any

combination of various types of data storage and/or transmission media that reside at one or more physical locations. Further, the I/O interfaces can comprise any system for exchanging information with one or more external device. Still further, it is understood that one or more additional components (e.g., system software, math co-processing unit, etc.) can be included in the hardware.

While shown and described herein as virtualization in a multi-core processor, it is understood that the invention further provides various alternative embodiments. For example, in one embodiment, the invention provides a computer-readable/useable medium that includes computer program code to enable a computer infrastructure to provide visualization in a multi-core processor. To this extent, the computer-readable/useable medium includes program code that implements the process(es) of the invention. It is understood that the terms computer-readable medium or computer useable medium comprises one or more of any type of physical embodiment of the program code. In particular, the computer-readable/useable medium can comprise program code embodied on one or more portable storage articles of manufacture (e.g., a compact disc, a magnetic disk, a tape, etc.), on one or more data storage portions of a computing device (e.g., a fixed disk, a read-only memory, a random access memory, a cache memory, etc.), and/or as a data signal (e.g., a propagated signal) traveling over a network (e.g., during a wired/wireless electronic distribution of the program code).

In another embodiment, the invention provides a method (e.g., business) that performs the process of the invention on a subscription, advertising, and/or fee basis. That is, a service provider, such as a Solution Integrator, could offer to provide virtualization in a multi-core processor. In this case, the service provider can create, maintain, support, etc., a computer infrastructure, such as computer infrastructure that performs the process of the invention for one or more customers. In return, the service provider can receive payment from the customer(s) under a subscription and/or fee agreement and/or the service provider can receive payment from the sale of advertising content to one or more third parties.

In still another embodiment, the invention provides a processing method. In this case, a computer infrastructure can be provided and one or more systems for performing the process

of the invention can be obtained (e.g., created, purchased, used, modified, etc.) and deployed to the computer infrastructure. To this extent, the deployment of a system can comprise one or more of: (1) installing program code on a computing device from a computer-readable medium; (2) adding one or more computing devices to the computer infrastructure; and (3) incorporating and/or modifying one or more existing systems of the computer infrastructure to enable the computer infrastructure to perform the process of the invention.

As used herein, it is understood that the terms "program code" and "computer program code" are synonymous and mean any expression, in any language, code or notation, of a set of instructions intended to cause a computing device having an information processing capability to perform a particular function either directly or after either or both of the following: (a) conversion to another language, code or notation; and/or (b) reproduction in a different material form. To this extent, program code can be embodied as one or more of: an application/software program, component software/a library of functions, an operating system, a basic I/O system/driver for a particular computing and/or I/O device, and the like.

A data processing system suitable for storing and/or executing program code can be provided hereunder and can include at least one processor communicatively coupled, directly or indirectly, to memory element(s) through a system bus. The memory elements can include, but are not limited to, local memory employed during actual execution of the program code, bulk storage, and cache memories that provide temporary storage of at least some program code in order to reduce the number of times code must be retrieved from bulk storage during execution. Input/output or I/O devices (including, but not limited to, keyboards, displays, pointing devices, etc.) can be coupled to the system either directly or through intervening I/O controllers.

Network adapters also may be coupled to the system to enable the data processing system to become coupled to other data processing systems, remote printers, storage devices, and/or the like, through any combination of intervening private or public networks. Illustrative network adapters include, but are not limited to, modems, cable modems and Ethernet cards.

The foregoing description of various aspects of the invention has been presented for purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed, and obviously, many modifications and variations are possible. Such modifications and variations that may be apparent to a person skilled in the art are intended to be included within the scope of the invention as defined by the accompanying claims.

CLAIMS

1. A multi-core processor, comprising:

a set of main processing elements each comprising a set of virtualized control threads; and

a set of groups of sub-processing elements, each one of the set of virtualized control threads controlling at least one of the set of groups of sub-processing elements.

2. The multi-core processor of claim 1, each of the set of groups of sub-processing elements comprising a plurality of sub-processing elements.

3. The multi-core processor of claim 1 or claim 2, the set of virtualized control threads being configured to send program code and data to the set of groups of sub-processing elements.

4. The multi-core processor of claim 3, the set of virtualized control threads being further configured to collect computation results from the set of groups of sub-processing elements.

5. The multi-core processor of any preceding claim, the set of virtualized control threads controlling a clock speed, power consumption and computation loading of the set of groups of sub-processing elements.

6. The multi-core processor of any preceding claim, the set of virtualized control threads being configured to send and receive power control requests between the set of main processing elements and the set of groups of sub-processing elements.

7. The multi-core processor of any preceding claim, the set of main processing elements comprising a plurality of main processing elements.

8. The multi-core processor of any preceding claim, each one of the set of visualized control threads being associated with a single group of sub-processing elements.

9. A processing method, comprising:

associating a set of main processing elements with a set of groups of sub-processing elements using a set of virtualized control threads; and

controlling the set of groups of sub-processing elements using the set of virtualized control threads.

10. The processing method of claim 9, further comprising sending program code and data to the set of groups of sub-processing elements via the set of virtualized control threads.

11. The processing method of claim 9 or claim 10, further comprising receiving computation results from the set of groups of sub-processing elements via the set of virtualized control threads.

12. The processing method of any of claims 9 to 11, the controlling comprising controlling a clock speed, a power consumption and a computation loading of the set of groups of sub-processing elements.

13. The processing method of any of claims 9 to 12, the set of virtualized control threads being embodied as program code within the set of main processing elements.

14. The processing method of any of claims 9 to 13, the set of main processing elements, the set of groups of sub-processing elements, and the set of virtualized control threads comprising a multi-core processor.

15. A computer program comprising computer program code to, when loaded into a computer system and executed thereon, cause said computer system to perform all the steps of a method according to any of claims 9 to 14.

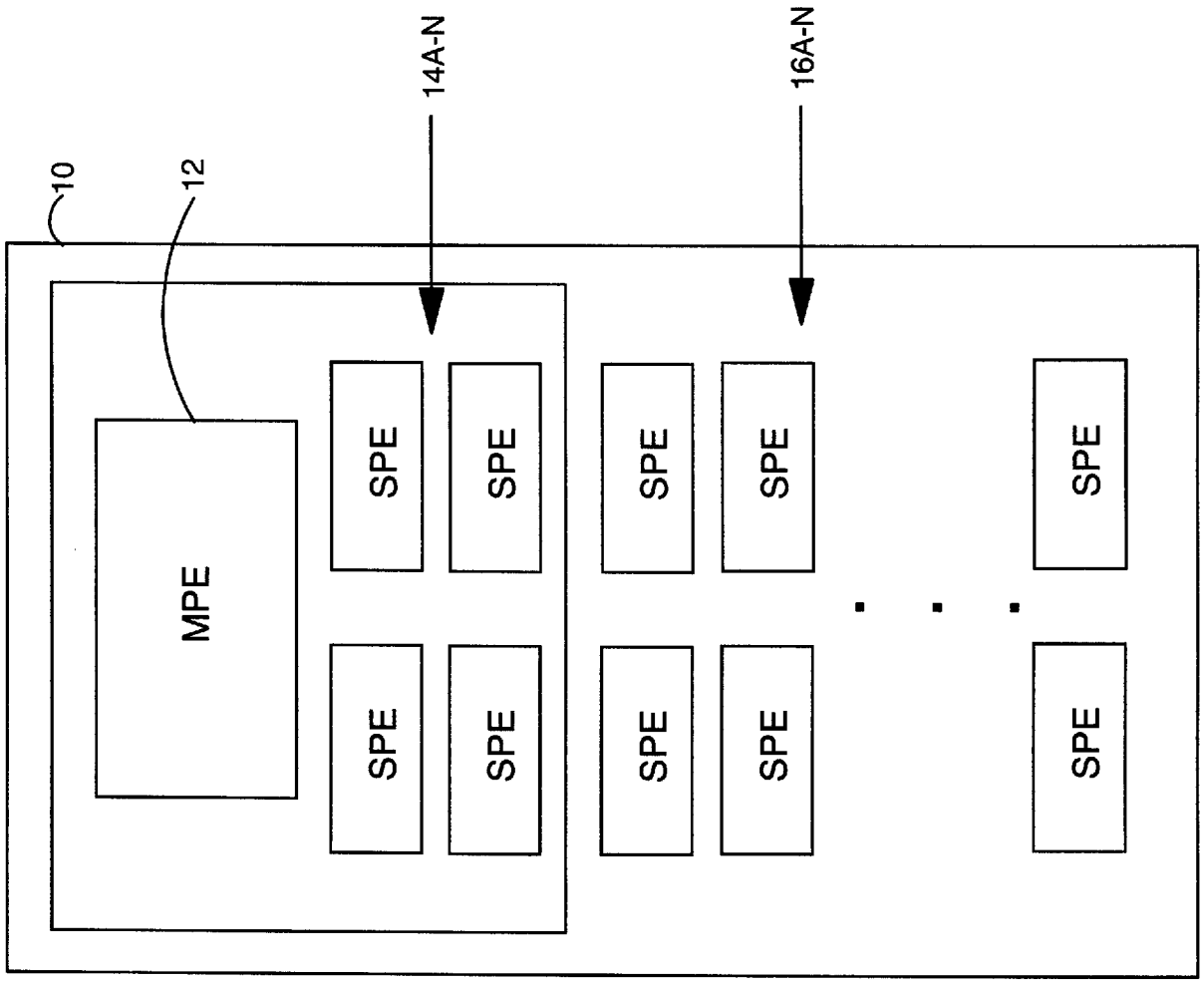

FIG. 1

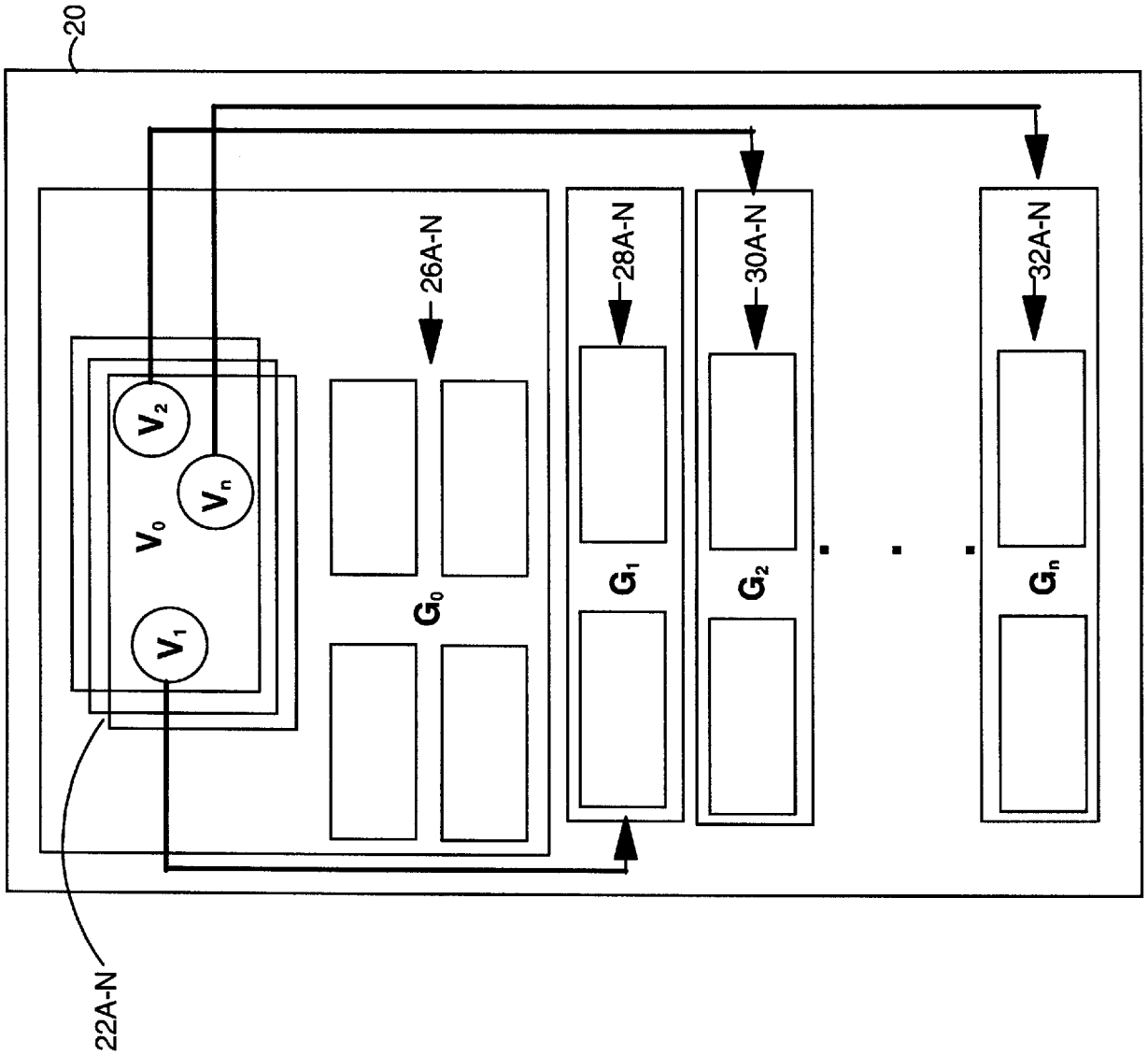

FIG. 2

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2009/061396

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F12/08 G06F13/364 G06F15/163

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

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 1 770 520 A2 (SONY COMP ENTERTAINMENT INC. [JP]) 4 April 2007 (2007-04-04) abstract paragraph [0008] - paragraph [0016] paragraph [0035] - paragraph [0042] -----	1-15
X	EP 1 416 377 A1 (ST MICROELECTRONICS INC [US]) 6 May 2004 (2004-05-06) abstract paragraph [0007] - paragraph [0008] paragraph [0013] - paragraph [0017] -----	1-15

-/--

[X] Further documents are listed in the continuation of Box C.

[X] 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 document 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	Date of mailing of the international search report
23 November 2009	04/12/2009

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

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2009/061396

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	KAHLE J A ET AL: "Introduction to the Cell multiprocessor" IBM JOURNAL OF RESEARCH AND DEVELOPMENT, INTERNATIONAL BUSINESS MACHINES CORPORATION, NEW YORK, NY, US, vol. 49, no. 4-5, 1 July 2005 (2005-07-01) , pages 589-604, XP002415988 ISSN: 0018-8646 abstract page 591, right-hand column, line 9 - page 596, right-hand column, line 5	1-15

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2009/061396

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 1770520 A2	04-04-2007	CN 1941780 A	04-04-2007
		CN 1972293 A	30-05-2007
		JP 2007095065 A	12-04-2007
		US 2007074206 A1	29-03-2007
EP 1416377 A1	06-05-2004	JP 2004152305 A	27-05-2004
		US 2004088519 A1	06-05-2004