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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
9 August 2012 (09.08.2012)

WIPO | PCT

(10) International Publication Number
WO 2012/104674 A1

(51) International Patent Classification:
G06F 9/06 (2006.01) *G06F 9/30* (2006.01)

(21) International Application Number:
PCT/IB2011/050419

(22) International Filing Date:
31 January 2011 (31.01.2011)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant *(for all designated States except US)*: **FREESCALE SEMICONDUCTOR, INC.** [US/US]; 6501 William Cannon Drive West, Austin, Texas 78735 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **MOSKOVICH, Ilia** [IL/IL]; HaKomemiut 37/14, 82091 Kiriat-Gat (IL). **AMIR, Aviram** [IL/IL]; 12 Moshe Sne, 49725 Petach-Tiqwa (IL). **BARAK, Itzhak** [IL/IL]; Ha-Pa'amon 12, 60920 Kadima (IL). **BEN ZEEV, Eliezer** [IL/IL]; Ben Gurion rd., 138-932, 59301 Bat Yam (IL).

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

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

(54) Title: INTEGRATED CIRCUIT DEVICE AND METHOD FOR DETERMINING AN INDEX OF AN EXTREME VALUE WITHIN AN ARRAY OF VALUES

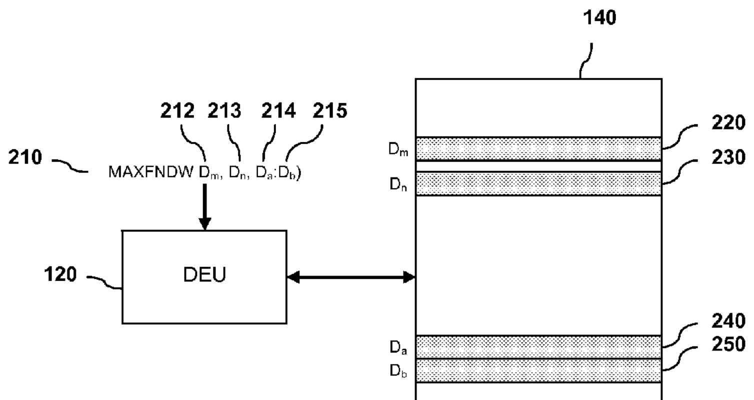


FIG. 2

(57) **Abstract**: An integrated circuit device (105) comprises at least one digital signal processor, DSP, module (100), the at least one DSP module (100) comprising a plurality of data registers (140) and at least one data execution unit, DEU, module (120) arranged to execute operations on data stored within the data registers (140). The at least one DEU module (120) is arranged to, in response to receiving an extreme value index instruction (210), compare a previous extreme value (425) located within a first data register set (220, 230) of the DSP module (100) with at least one input vector data value (440, 445, 450, 455) located within a second data register set (240, 250) of the DSP module (100), and determine an extreme value thereof. The at least one DEU module (120) is further arranged to, if the determined extreme value comprises an input vector data value (440, 445, 450, 455) located within the second data register set (240, 250), store the determined extreme value in the first data register set (220, 230), determine an index value for the determined extreme value, and store the determined index value in the first data register set (220, 230).

Title: INTEGRATED CIRCUIT DEVICE AND METHOD FOR DETERMINING AN INDEX OF AN EXTREME VALUE WITHIN AN ARRAY OF VALUES

Description

Field of the invention

The field of this invention relates to an integrated circuit device and method for determining an index of an extreme value within an array of values.

Background of the invention

Digital Signal Processor (DSP) applications are becoming increasingly more demanding in their requirements of DSP cores. For example, the increased data rates specified for the fourth generation (4G) of mobile telecommunication systems require significantly higher DSP performance. One operation that is often required to be performed is that of finding the index of an element comprising, say, a maximum or minimum value, or modulus (absolute maximum) value, within a given systematic arrangement of values such as an array (as opposed to the maximum/minimum value itself). For clarity, an index comprises an identifier for a specific element within, say, an array, such as in the form of a relative location of the array element within the array. For example, the search for the index of a maximum value within an array is often used within communication systems for such algorithms as timing/frequency offset detection (whereby a received signal is correlated with a pilot sequence, and the index of the maximum correlation typically corresponds to the timing/frequency offset), as well as for finding one of many possible transmitted sequences using cross correlation e.g. a maximum likelihood (ML) decoder.

In order to perform such max/min value index finding operations using traditional DSP implementations, it is typically necessary for the application program code to comprise an 'if' statement that results in a 'change of flow' in the application program code. To improve the overall throughput of DSPs, a pipelined architecture is typically implemented in order to reduce the overall number of cycles per instruction, and thereby improve the throughput of the DSP. However, a change of flow, such as may result from the use of an 'if' statement, effectively voids any benefit provided by such a pipelined architecture for that part of the application program code, significantly reducing the efficiency of the DSP. In addition, the use of an 'if' statement to perform such max/min value index finding operations prevents more than one max/min value index finding operation being performed per cycle (due to the limitation of using a dedicated conditional flag within the DSP), irrespective of the ability of the processing parallelism within DSP architecture.

Since such max/min value index finding operations may be required to be performed on a regular basis within a mobile telecommunication receiver, such a reduction in the efficiency of the DSP for performing such operations can have a significant detrimental effect on the overall performance of the DSP.

Summary of the invention

The present invention provides an integrated circuit device and method for determining an index of an extreme value within an array of values, as described in the accompanying claims.

Specific embodiments of the invention are set forth in the dependent claims.

These and other aspects of the invention will be apparent from and elucidated with reference to the embodiments described hereinafter.

Brief description of the drawings

Further details, aspects and embodiments of the invention will be described, by way of example only, with reference to the drawings. In the drawings, like reference numbers are used to identify like or functionally similar elements. Elements in the figures are illustrated for simplicity and clarity and have not necessarily been drawn to scale.

FIG. 1 illustrates a simplified block diagram of an example of part of a DSP module.

FIG. 2 illustrates a simplified example of a data execution unit (DEU) module executing an extreme value index instruction.

FIG. 3 illustrates a simplified example of a one dimensional array of values.

FIG. 4 illustrates a simplified example of an execution of a maximum value index instruction.

FIG. 5 illustrates a simplified flowchart of an example of a method for determining an index for an extreme value within an array of values.

Detailed description

Examples of the present invention will now be described with reference to an example of a digital signal processor (DSP) architecture. However, it will be appreciated that the present invention is not limited to the specific DSP architecture herein described with reference to the accompanying drawings, and may equally be applied to alternative DSP architectures. For the illustrated example, a DSP architecture is provided comprising separate data and address registers. Alternatively, the address registers may be combined with the data registers, and the combined registers being used to provide address storage, etc. Furthermore, for the illustrated examples, the DSP architecture is shown as comprising four data execution units. However, it will be appreciated that examples of the present invention may equally be implemented within a DSP architecture comprising any other number of data execution units, e.g. one, two, eight, etc. Additionally, because the illustrated example embodiments of the present invention may for the most part, be implemented using electronic components and circuits known to those skilled in the art, details will not be explained in any greater extent than that considered necessary as illustrated below, for the understanding and appreciation of the underlying concepts of the present invention and in order not to obfuscate or distract from the teachings of the present invention.

Referring first to FIG. 1, there is illustrated a simplified block diagram of an example of a part of a DSP module 100 in accordance with some example embodiments of the present invention. In the illustrated example, the DSP module 100 is a part of an integrated circuit device, illustrated generally at 105, and comprises at least one program control unit (PCU) 110, one or more data execution unit (DEU) modules 120, at least one address generation unit (AGU) 130 and a plurality of data registers, illustrated generally at 140. The PCU 110 is arranged to receive DSP instructions, as illustrated generally at 115, and to cause an execution of operations within the DSP module 100 in accordance with the received DSP instructions. For example, the PCU 110 may receive a DSP instruction, for example stored within an instruction buffer (not shown), where the received DSP instruction requires one or more operations to be performed on one or more bits/bytes/words/etc. of data. A data 'bit' typically refers to a single unit of binary data comprising either a logic '1' or logic '0', whilst a 'byte; typically refers to a block of 8-bits. A data 'word' may comprise one or more bytes of data, for example two bytes (16-bits) of data, depending on the particular DSP architecture. Upon receipt of such a DSP instruction, the PCU 110 generates and outputs instructions and/or control signals to the various other components within the DSP module 100 in order for the required operations to be performed. The AGU 130 is arranged to generate address values for accessing system memory, and may comprise one or more address registers as illustrated generally at 135. The data registers 140 provide storage for fetched from system memory, and on which one or more operations is/are to be performed, and from which data may be written to system memory. The DEU modules 120 are arranged to perform operations on data (either provided directly thereto or stored within the data registers 140) in accordance with instructions and control signals received from the PCU 110. As such, the DEU modules 120 may comprise arithmetic logic units (ALUs), etc.

As previously mentioned, DSP applications may require an <u>index</u> of a data element comprising a maximum or minimum value or modulus (absolute maximum) value (hereinafter generally referred to as an extreme value) within a given systematic arrangement of values such as an array (as opposed to the maximum/minimum value itself). In accordance with some example embodiments of the present invention, at least one of the DEU modules 120 is arranged to, upon receipt of an extreme (e.g. maximum or minimum) value index instruction, compare a previous extreme value located within a first data register set of the DSP module 100 to at least one input vector data value located within a second data register set of the DSP module 100, and determine an extreme value thereof. The at least one DEU module 120 is further arranged to, if the determined extreme value comprises an input vector data value located within the second data register set, store the determined extreme value in the first data register set. The at least one DEU module 120 also determines an index value for the determined extreme value, and stores the determined index value in the first data register set.

FIG. 2 illustrates a simplified example of a DEU module 120 executing such an extreme value index instruction. For the example illustrated in FIG. 2, the DEU module 120 receives a maximum value index instruction 210, for example from the PCU module 110 of FIG. 1. The maximum value index instruction 210 comprises, in one example, one or more data register identifiers for one or more data registers that make up a first data register set. For the illustrated example, the first register set comprises a first data register identifier 212 for a first data register

220, and a second data register identifier 213 for a second data register 230. The maximum value index instruction 210 further comprises one or more data register identifiers for one or more further data registers that make up a second data register set, which for the illustrated example comprises a third data register identifier 214 for a third data register 240, and a fourth data register identifier 215 for a fourth data register 250. Upon receipt of the maximum value index instruction 210, the DEU module 120 compares a previous extreme value located within the first data register set (i.e. within one of the first or second data registers 220, 230 for the illustrated example) to one or more input vector values located within the second data register set (i.e. within the third and fourth data registers 240, 250 for the illustrated example), and determines a maximum value therefor. If the determined maximum value comprises an input vector data value located within the second data register set (i.e. within one of the third and fourth data registers 240, 250 for the illustrated example), the DEU module 120 stores the determined maximum value in the first data register set (i.e. within one of the first or second data registers 240, 250 for the illustrated example). Furthermore, the DEU module 120, in this example, determines an index value for the determined maximum value, and also stores the determined index value in the first data register set (i.e. within one of the first or second data registers 240, 250 for the illustrated example).

In this manner, a current maximum value from previously performed input vector value comparisons, and an index value therefor, are both stored in a set of data registers separate from the input vector values from which the index of the maximum value is being sought. In this manner, for each execution cycle of the DSP module 100, a new input vector (containing values from an array, or other systematic arrangement, of values for which the index of a maximum value is required) may be loaded into the second set of data registers (e.g. data registers 240, 250), whilst data within the first set of data registers (e.g. data registers 220, 230) is maintained between execution cycles. Thus, in this manner, input vectors comprising data values from the array of values may be sequentially loaded into the second set of data registers in, for example, substantially consecutive execution cycles of the DSP module 100, with the DEU module 120 updating the current maximum value and index value therefor within the first set of data registers.

For example, FIG. 3 illustrates an array 300 of values. For ease of understanding, array 300 is a one dimensional array comprising a plurality of array elements 310; each array element 310 containing a data value. The array 300 may be partitioned in to **n** vectors, as illustrated at 320, 322, 324, 326, 328. As described above, each vector 320, 322, 324, 326, 328 may be sequentially loaded into the second set of data registers (e.g. data registers 240, 250 in FIG. 2) in order to determine the index of the maximum value within the array 300. More specifically, and as illustrated in FIG. 3, within each execution cycle 330, 332, 334, 336, 338 of the DEU 120, a vector 320, 322, 324, 326, 328 comprising data values from the array 300 may be sequentially loaded into the second set of data registers 240, 250, and the maximum value index instruction 210 executed to aggregately determine the maximum value within the loaded vectors 320, 322, 324, 326, 328 and index value therefor. Accordingly, the DEU module 120 is capable of processing (i.e. determining the max/min value and an index value therefore) **i** values from an array of values each execution cycle, where **i** equals the number of values that may be stored within the second set of data registers (and thus the number of values within each vector 320, 322, 324, 326, 328). Furthermore, since the current maximum value and index value therefor may be maintained and updated substantially indefinitely within the first data register set, whilst input vectors 320, 322, 324, 326, 328 comprising data values are sequentially loaded into the second set of data registers in consecutive execution cycles, an index value for a maximum value within substantially any size of array may be determined without any additional processing overhead beyond that of executing consecutive maximum value index instructions 210. That is to say, for an array of **j** values, the number of execution cycles **n** required for a single DEU module 120 to find the index of the maximum value will be **j/i**. The above example may apply equally to the execution of a minimum value index instruction. Thus, in this manner, the example of an extreme value index instruction described above provides an efficient mechanism for finding the index of a maximum or minimum value within an array.

Furthermore, in some examples, the ability of the DEU modules 120 within the DSP module 100 to execute such an extreme value index instruction may substantially alleviate a need for an equivalent operation to be implemented using conventional application program code instructions, and thereby alleviate the necessity of using ‘if’ statements that result in changes of program flow. As such, the use of such an extreme value index instruction may enable some of the benefits of a pipelined architecture to be fully exploited, along with any parallel processing ability provided within DSP architecture. For example, extreme value index instructions may be provided to a plurality of DEU modules 120. In this manner, one or more DEU modules 120 may perform the operations of finding indexes for maximum and/or minimum values within multiple arrays in parallel. Alternatively, the operation of finding an index for a maximum/minimum value within a single array may be divided between multiple DEU modules 120.

Referring now to FIG. 4, there is illustrated a simplified example of a maximum value index instruction being executed according to some example embodiments of the present invention, for example as may be implement by one or more of the DEU modules 120. As for the example illustrated in FIG. 2, a first set of data registers comprises a first data register 220 and a second data register 230. Each data register may comprise, for example, a 32-bit data register divided into two separately accessible 16-bit sections. Thus, for the illustrated example a current maximum value 425, for example from previous executions of the maximum value index instruction, is stored within a first section of the first data register 220. An index value 430 for this current maximum value 425 is stored within, for the illustrated example, a first section of the second data register 230. A second set of data registers comprises a third data register 240 and a fourth data register 250. A new input vector (containing values 440, 445, 450, 455 from an array, or other systematic arrangement, of values for which the index of, in the illustrated example, a maximum value is required) may be loaded into the second set of data registers (i.e. data registers 240, 250 for the illustrated example) in each execution cycle. A comparison of the current maximum value 425 located within the first data register set 220, 230 and the input vector data values 440, 445, 450, 455 located within a second data register set 240, 250 is made, for example by using a first combination logic unit illustrated generally at 410, and where a maximum value is determined thereby.

If the determined maximum value comprises one of the input vector data values 440, 445, 450, 455 located within the second data register set 240, 250, the determined maximum value may be stored within the first set of data registers 220, 230. Specifically for the illustrated example, if the determined maximum value comprises one of the input vector data values 440, 445, 450, 455 located within the second data register set 240, 250, the previous maximum value 425 located within the first section of the first data register 220 may be overwritten with the (new) determined maximum value. Furthermore, an index value for the new maximum value may be determined, and also stored within the first set of data registers 220, 230. Specifically for the illustrated example, if the determined maximum value comprises one of the input vector data values 440, 445, 450, 455 located within the second data register set 240, 250, the index 430 of the previous maximum value 425 located within the first section of the second data register 230 may be overwritten with the index value for the new maximum value.

In some examples, an index value for the new maximum value may be based on both a current index value for the current input vector stored within the second set of data registers 240, 250 and an index value for a location of the new maximum value within the second data register set 240, 250. For example, and as illustrated in FIG. 4, a current index value 435 for the current input vector within the second set of data registers 240, 250 may be located within the first set of data registers, and specifically for the illustrated example within a second section of the second data register 230. Such an current index value 435 for the current input vector may be provided to second combination logic unit, such as illustrated generally at 460. Furthermore, the first combination logic unit 410 arranged to determine the maximum value may be arranged to provide an index value 415, or other identifier, for the location of the determined maximum value. In this manner, if the determined maximum value comprises one of the input vector data values 440, 445, 450, 455 located within the second data register set 240, 250, the second combination logic unit 460 may add, or otherwise combine, the index value 435 for the current input vector and the index value 415 for the location of the determined maximum value within the input vector in order to generate an index value 430 for the new maximum value 425. This index value 430 is then stored within the first data register set 220, 230.

In order for the current index value 435 to be accurate, in some examples, it may be necessary for the current index value 435 to be updated each time a new input vector is loaded into the second data register set 240, 250. Accordingly, in some examples, during execution of each maximum value index instruction, the current index value may be, say, incremented by a defined amount. For the illustrated example, an index stride value 420 is located within the first data register set 220, 230, and in particular for the illustrated example is stored within a second section of the first data register 220. The current index value 435 for the current input vector is incremented during each execution of a maximum value index instruction by an amount equal to the index stride value 420, as illustrated generally at 470. The incremented current index value may then be stored back in the first data register set 220, 230, overwriting the previous current index value 435. In this manner, the current index value 435 is incremented to a new value ready for the execution of the maximum value index instruction with the next input vector.

Referring back to the comparison of the current maximum value 425 located within the first data register set 220, 230 and the input vector data values 440, 445, 450, 455 located within the second data register set 240, 250, by first combination logic unit 410, if the determined maximum value comprises the current maximum value 425 located within the first data register set 220, 230, then the current maximum value 425 is to be maintained. For example, no value need be stored in the first data register set 220, 230 since the correct value is already present. However, to simplify the implementation in one example, the first combination logic unit 410 may be arranged to store whatever maximum value is determined. As such, if the determined maximum value comprises the current maximum value 425 located within the first data register set 220, 230, then the current maximum value 425 may be re-written over itself in the first data register set 220, 230. Similarly, if the determined maximum value comprises the current maximum value 425 located within the first data register set 220, 230, then the second combination logic unit 460 for determining an index value for the maximum value may be arranged to either store no new value within the first data register set 220, 230, or to re-write the index value 430 of the current maximum value 425 over itself.

FIG.3 illustrates an example of the execution of a maximum value index instruction. The execution may also be performed for a minimum value index instruction, with the exception of an equivalent for the first combination logic unit 410 being arranged to perform a comparison of a current value 425 located within the first data register set 220, 230 and the input vector data values 440, 445, 450, 455 located within the second data register set 240, 250, in order to determine a minimum value.

Referring now to FIG. 5, there is illustrated a simplified flowchart 500 of an example of a method for determining an index for an extreme value within an array of values according to some example embodiments of the present invention, for example as may be implemented within the DSP module 100 of FIG. 1. The method starts at 505, and moves on to 510 where operating data is initialised within a first set of data registers, for example in response to an initialisation instruction. For example, for the embodiment illustrated in FIG. 4, such operating data may comprise the index stride value 420, a current maximum value 425, an index value for the current maximum value 430, and a current index value for input vector 435. The initialisation of such operating data may comprise setting each value to a default or definable initial value. At 515, input vector data is loaded into a second set of data registers (which in some examples contains values from an array of values for which the index of a maximum value is required). A comparison of the input vector data values (stored within the second set of data registers) and the current maximum value (stored within the first set of data registers) is then performed at 520, in order to determine a maximum value there between. For the illustrated example, if the determined maximum value is one of the input vector data values located within the second data register set (i.e. a new maximum value), at 525, the method moves on to 530, where the previous current maximum value stored within the first data register set is replaced with the new maximum value. An index for the new maximum value is then determined at 535, and an index value for the previous current maximum value within the first data register set is replaced with the index value for the new current maximum value at 540. A current index value for the input vector within the second data register set is then incremented, for example by an amount equal to an index stride (such as index stride 420) located within the first data register set, at 545. If more data values remain within the array of values for which the index of a maximum value is required, at 550, the method loops back to 515, and the next input vector of values from the array is loaded into the second data register set. However, if no further data values remain, the method ends at 555. Referring back to 525, for the illustrated example, if the determined maximum value comprises the current maximum value 425 located within the first data register set, the method jumps to 545, thereby maintaining the current maximum value and the index value therefor within the first data register set. However, in some examples, if the determined maximum value is equal to the current maximum value 425 located within the first data register set, the method may alternatively comprise re-writing the current maximum value and/or the index therefor into the first data register set. The current index for the input vector within the second data register is incremented 545, and the method either loops back to 515 at 550, in order to load the next input vector, or ends at 555.

In the foregoing specification, the invention has been described with reference to specific examples of embodiments of the invention. It will, however, be evident that various modifications and changes may be made therein without departing from the broader spirit and scope of the invention as set forth in the appended claims.

The connections as discussed herein may be any type of connection suitable to transfer signals from or to the respective nodes, units or devices, for example via intermediate devices. Accordingly, unless implied or stated otherwise, the connections may for example be direct connections or indirect connections. The connections may be illustrated or described in reference to being a single connection, a plurality of connections, unidirectional connections, or bidirectional connections. However, different embodiments may vary the implementation of the connections. For example, separate unidirectional connections may be used rather than bidirectional connections and vice versa. Also, plurality of connections may be replaced with a single connection that transfers multiple signals serially or in a time multiplexed manner. Likewise, single connections carrying multiple signals may be separated out into various different connections carrying subsets of these signals. Therefore, many options exist for transferring signals.

Although specific conductivity types or polarity of potentials have been described in the examples, it will be appreciated that conductivity types and polarities of potentials may be reversed.

Each signal described herein may be designed as positive or negative logic. In the case of a negative logic signal, the signal is active low where the logically true state corresponds to a logic level zero. In the case of a positive logic signal, the signal is active high where the logically true state corresponds to a logic level one. Note that any of the signals described herein can be designed as either negative or positive logic signals. Therefore, in alternate embodiments, those signals described as positive logic signals may be implemented as negative logic signals, and those signals described as negative logic signals may be implemented as positive logic signals.

Furthermore, the terms "assert" or “set” and "negate" (or "de-assert" or “clear”) are used herein when referring to the rendering of a signal, status bit, or similar apparatus into its logically true or logically false state, respectively. If the logically true state is a logic level one, the logically false state is a logic level zero. And if the logically true state is a logic level zero, the logically false state is a logic level one.

Those skilled in the art will recognize that the boundaries between logic blocks are merely illustrative and that alternative embodiments may merge logic blocks or circuit elements or impose an alternate decomposition of functionality upon various logic blocks or circuit elements. Thus, it is to be understood that the architectures depicted herein are merely exemplary, and that in fact many other architectures can be implemented which achieve the same functionality. For example, for clarity the PCU module 110, AGU 130, DEU modules 120 and data registers 140 have been illustrated as comprising substantially discrete logical elements within the DSP module 100. However it will be appreciated that such functional elements may be provided within any suitable distribution of physical modules.

Any arrangement of components to achieve the same functionality is effectively "associated" such that the desired functionality is achieved. Hence, any two components herein combined to achieve a particular functionality can be seen as "associated with" each other such that the desired functionality is achieved, irrespective of architectures or intermediary components. Likewise, any two components so associated can also be viewed as being "operably connected," or "operably coupled," to each other to achieve the desired functionality.

Furthermore, those skilled in the art will recognize that boundaries between the above described operations merely illustrative. The multiple operations may be combined into a single operation, a single operation may be distributed in additional operations and operations may be executed at least partially overlapping in time. Moreover, alternative embodiments may include multiple instances of a particular operation, and the order of operations may be altered in various other embodiments.

Also for example, the examples, or portions thereof, may implemented as soft or code representations of physical circuitry or of logical representations convertible into physical circuitry, such as in a hardware description language of any appropriate type.

Also, the invention is not limited to physical devices or units implemented in non-programmable hardware but can also be applied in programmable devices or units able to perform the desired device functions by operating in accordance with suitable program code, such as mainframes, minicomputers, servers, workstations, personal computers, notepads, personal digital assistants, electronic games, automotive and other embedded systems, cell phones and various other wireless devices, commonly denoted in this application as ‘computer systems’.

However, other modifications, variations and alternatives are also possible. The specifications and drawings are, accordingly, to be regarded in an illustrative rather than in a restrictive sense.

In the claims, any reference signs placed between parentheses shall not be construed as limiting the claim. The word ‘comprising’ does not exclude the presence of other elements or steps then those listed in a claim. Furthermore, the terms “a” or “an”, as used herein, are defined as one or more than one. Also, the use of introductory phrases such as “at least one” and “one or more” in the claims should not be construed to imply that the introduction of another claim element by the indefinite articles "a" or "an" limits any particular claim containing such introduced claim element to inventions containing only one such element, even when the same claim includes the introductory phrases "one or more" or "at least one" and indefinite articles such as "a" or "an". The same holds true for the use of definite articles. Unless stated otherwise, terms such as “first” and “second” are used to arbitrarily distinguish between the elements such terms describe. Thus, these terms are not necessarily intended to indicate temporal or other prioritization of such elements. The mere fact that certain measures are recited in mutually different claims does not indicate that a combination of these measures cannot be used to advantage.

Claims

1. An integrated circuit device (105) comprising at least one digital signal processor, DSP, module (100), the at least one DSP module (100) comprising a plurality of data registers (140) and at least one data execution unit, DEU, module (120) arranged to execute operations on data stored within the data registers (140);

wherein the at least one DEU module (120) is arranged to, in response to receiving an extreme value index instruction (210):

compare a previous extreme value (425) located within a first data register set (220, 230) of the DSP module (100) with at least one input vector data value (440, 445, 450, 455) located within a second data register set (240, 250) of the DSP module (100); and

determine an extreme value thereof;

wherein the at least one DEU module (120) is further arranged to, if the determined extreme value comprises an input vector data value (440, 445, 450, 455) located within the second data register set (240, 250):

store the determined extreme value in the first data register set (220, 230);

determine an index value for the determined extreme value; and

store the determined index value in the first data register set (220, 230).

2. The integrated circuit device (105) of Claim 1 wherein the at least one DEU module (120) is arranged to overwrite the previous extreme value (425) located within the first data register set (220, 230) with the determined extreme value, if the determined extreme value comprises an input vector data value (440, 445, 450, 455) located within the second data register set (240, 250).

3. The integrated circuit device (105) of Claim 1 or Claim 2 wherein the at least one DEU module (120) is arranged to overwrite an index value (430) for the previous extreme value (425) located within the first data register set (220, 230) with the determined index value for the determined extreme value, if the determined extreme value comprises an input vector data value (440, 445, 450, 455) located within the second data register set (240, 250).

4. The integrated circuit device (105) of any preceding Claim wherein the at least one DEU module (120) is arranged to determine an index value therefor based on a current index value (435) located within the first data register set (220, 230) and an index value (415) for a location of the determined extreme value within the second data register set (240, 250), if the determined extreme value comprises an input vector data value (440, 445, 450, 455) located within the second data register set (240, 250).

5. The integrated circuit device (105) of Claim 5 wherein the at least one DEU module (120) is further arranged to increment the current index value (435) located within the first data register set (220, 230) during execution of the received extreme value index instruction (210).

6. The integrated circuit device (105) of Claim 5 wherein the at least one DEU module (120) is arranged to increment the current index value (435) by an amount equal to an index stride value (420) located within the first data register set (220, 230) during execution of the received extreme value index instruction (210),.

7. The integrated circuit device (105) of any preceding Claim wherein the first data register set comprises at least one data register (220, 230) arranged to store therein at least one from a group consisting of:

an index stride value (420);
a previous extreme value (425);
an index value for the previous extreme value (430); and
a current index value (435).

8. The integrated circuit device (105) of any preceding Claim wherein the extreme value index instruction (210) comprises a maximum value index instruction or a minimum value index instruction.

9. A method (500) for determining an index for an extreme value within an array of values; the method comprising:

comparing a previous extreme value located within a first data register set and at least one input vector data value located within a second data register set and determining an extreme value thereof (520); and

wherein the method further comprises, in response to determining that the extreme value thereof comprises an input vector data value located within the second data register set:

storing the determined extreme value in the first data register set (530);
determining an index value for the determined extreme value (535); and
storing the determined index value in the first data register set (540).

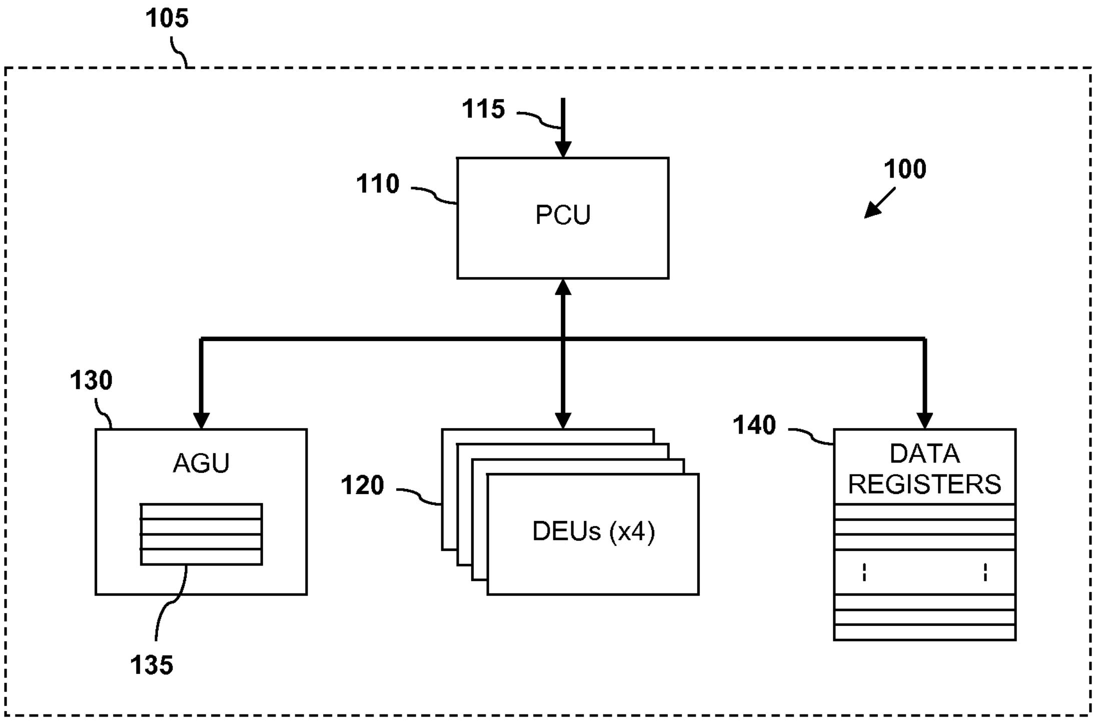

105
115
110
PCU
100
130
AGU
120
DEUs (x4)
140
DATA REGISTERS
135

FIG. 1

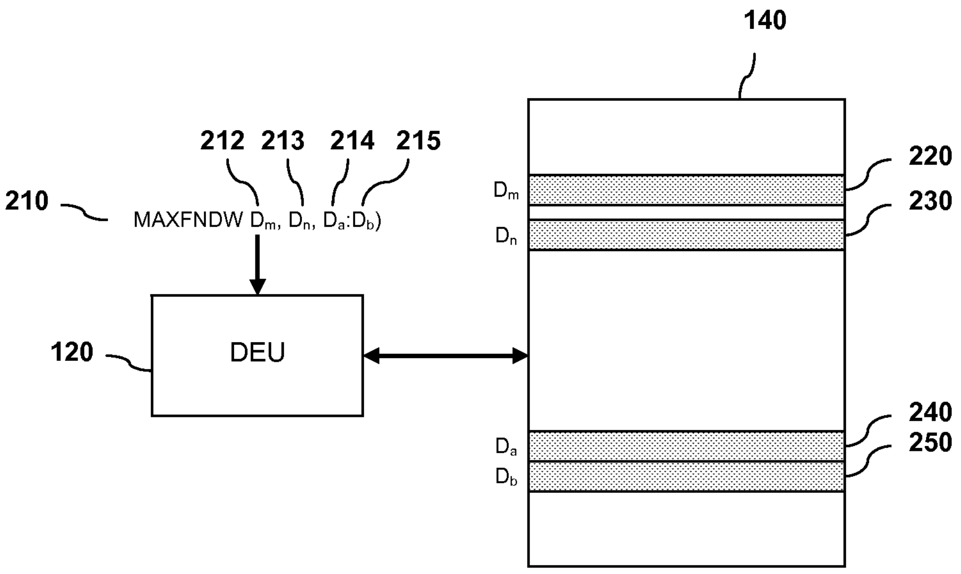
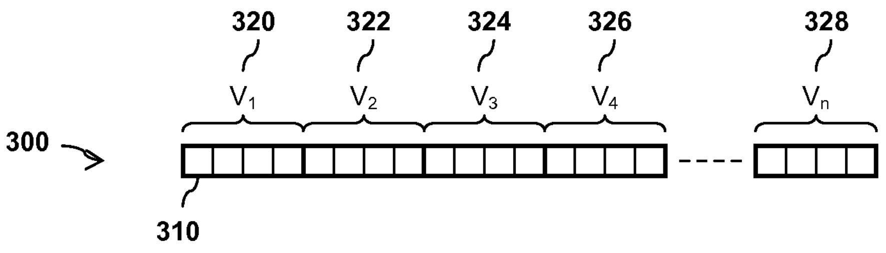

140
212 213 214 215
210
MAXFNDW D_m, D_n, D_a:D_b)
220
230
D_m
D_n
120
DEU
240
250
D_a
D_b

FIG. 2



330 Cycle$_1$ { **Ld V_1 to D_a:D_b** {Vector V_1 loaded into D_a:D_b}
MAXFNDW D_m, D_n, D_a:D_b) {Max value & index for V_1 updated in D_m:D_n}

332 Cycle$_2$ { **Ld V_2 to D_a:D_b** {Vector V_2 loaded into D_a:D_b}
MAXFNDW D_m, D_n, D_a:D_b) {Max value & index for V_1 and V_2 updated in D_m:D_n}

334 Cycle$_3$ { **Ld V_3 to D_a:D_b** {Vector V_3 loaded into D_a:D_b}
MAXFNDW D_m, D_n, D_a:D_b) {Max value & index for V_1 to V_3 updated in D_m:D_n}

336 Cycle$_4$ { **Ld V_4 to D_a:D_b** {Vector V_4 loaded into D_a:D_b}
MAXFNDW D_m, D_n, D_a:D_b) {Max value & index for V_1 to V_4 updated in D_m:D_n}

338 Cycle$_n$ { **Ld V_n to D_a:D_b** {Vector V_4 loaded into D_a:D_b}
MAXFNDW D_m, D_n, D_a:D_b) {Max value & index for V_1 to V_n updated in D_m:D_n}

FIG. 3

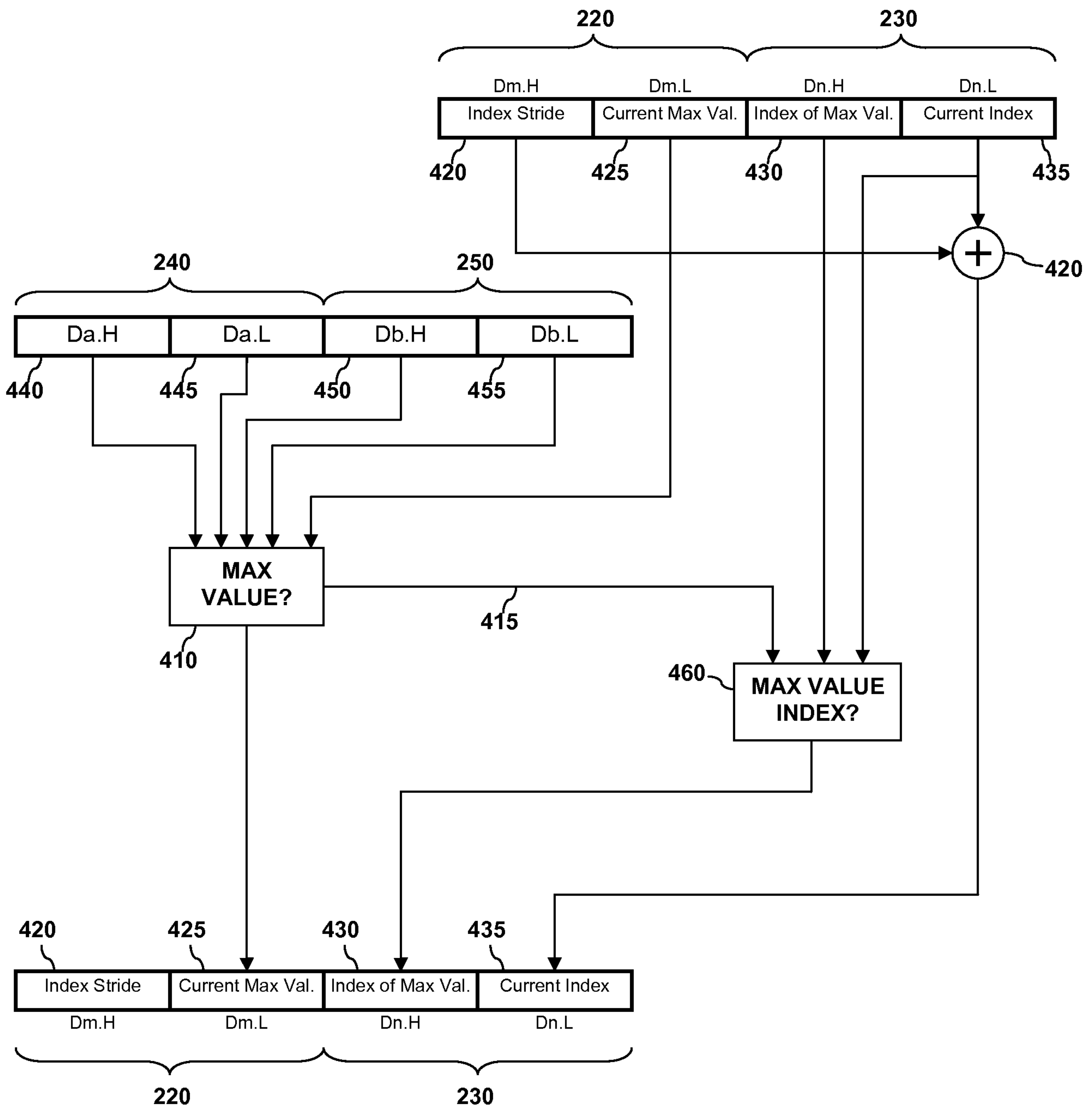

220
230
Dm.H
Dm.L
Dn.H
Dn.L
Index Stride
Current Max Val.
Index of Max Val.
Current Index
420
425
430
435
420
240
250
Da.H
Da.L
Db.H
Db.L
440
445
450
455
MAX VALUE?
410
415
460
MAX VALUE INDEX?
420
425
430
435
Index Stride
Current Max Val.
Index of Max Val.
Current Index
Dm.H
Dm.L
Dn.H
Dn.L
220
230

FIG. 4

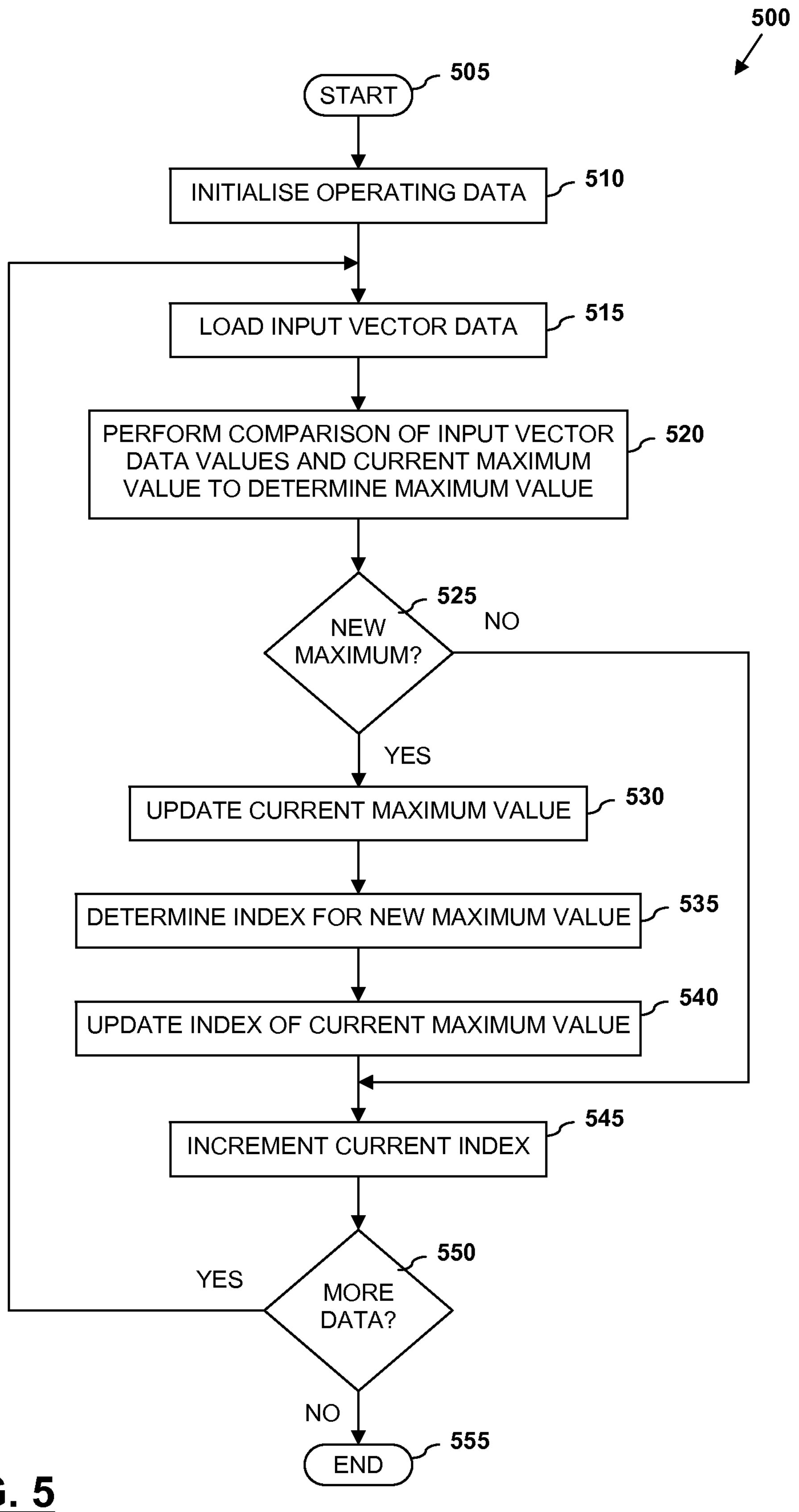

500
START
505
INITIALISE OPERATING DATA
510
LOAD INPUT VECTOR DATA
515
PERFORM COMPARISON OF INPUT VECTOR DATA VALUES AND CURRENT MAXIMUM VALUE TO DETERMINE MAXIMUM VALUE
520
NEW MAXIMUM?
525
NO
YES
UPDATE CURRENT MAXIMUM VALUE
530
DETERMINE INDEX FOR NEW MAXIMUM VALUE
535
UPDATE INDEX OF CURRENT MAXIMUM VALUE
540
INCREMENT CURRENT INDEX
545
MORE DATA?
550
YES
NO
END
555

FIG. 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IB2011/050419

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/06(2006.01)i, G06F 9/30(2006.01)i

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 9/06; G06F 15/00; G06F 17/00; G06F 17/30; G06F 7/00; G06F 7/06; G06F 11/00

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

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS(KIPO internal) & Keywords: digital signal processor, extreme value, array of value, data register set;

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2006-0101230 A1 (CHARLES P. ROTH et al.) 11 May 2006 See the abstract, claims 1-20 and figures 1-6.	1-9
A	US 2010-0057676 A1 (RAMAIYER KUMAR) 04 March 2010 See the abstract, claims 1-20 and figures 1-8.	1-9
A	US 2008-0201346 A1 (MONRO DONALD MARTIN) 21 August 2008 See the abstract, claims 1-63 and figures 1-8.	1-9
A	US 2007-0192655 A1 (THADI NAGARAJ) 16 August 2007 See the abstract, claims 1-50 and figures 1-6.	1-9

☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of 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
26 OCTOBER 2011 (26.10.2011)	**27 OCTOBER 2011 (27.10.2011)**

Name and mailing address of the ISA/KR	Authorized officer
Korean Intellectual Property Office Government Complex-Daejeon, 189 Cheongsa-ro, Seo-gu, Daejeon 302-701, Republic of Korea	BOK, Jin Yo 복진요
Facsimile No. 82-42-472-7140	Telephone No. 82-42-481-5113

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/IB2011/050419

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2006-0101230 A1	11.05.2006	CN 1230741 C	07.12.2005
		CN 1466715 A	07.01.2004
		CN 1766833 A	03.05.2006
		JP 2004-510245 T	02.04.2004
		JP 4380987 B2	09.12.2009
		TW 538349 B	21.06.2003
		US 6948056 B1	20.09.2005
		WO 02-27475 A2	04.04.2002
		WO 02-27475 A3	13.06.2002
US 2010-0057676 A1	04.03.2010	None	
US 2008-0201346 A1	21.08.2008	US 7707214 B2	27.04.2010
		WO 2008-103322 A2	28.08.2008
		WO 2008-103322 A3	26.02.2009
		WO 2008-103322 A3	28.08.2008
US 2007-0192655 A1	16.08.2007	CN 101379711 A	04.03.2009
		EP 1980023 A1	15.10.2008
		JP 2009-526445 A	16.07.2009
		JP 2009-526445 T	16.07.2009
		KR 10-1006032 B1	06.01.2011
		KR 2008-0099299 A	12.11.2008
		WO 2007-090204 A1	09.08.2007